(12)特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局

(43) 国際公開日
2001年11月1日 (01.11.2001)

PCT

(10) 国際公開番号
WO 01/82604 A1

(51) 国際特許分類[7]: H04N 5/91

(21) 国際出願番号: PCT/JP01/03413

(22) 国際出願日: 2001年4月20日 (20.04.2001)

(25) 国際出願の言語: 日本語

(26) 国際公開の言語: 日本語

(30) 優先権データ:
特願2000-183770 2000年4月21日 (21.04.2000) JP
特願2000-268042 2000年9月5日 (05.09.2000) JP

(71) 出願人 (米国を除く全ての指定国について): ソニー株式会社 (SONY CORPORATION) [JP/JP]; 〒141-0001 東京都品川区北品川6丁目7番35号 Tokyo (JP).

(72) 発明者; および
(75) 発明者/出願人 (米国についてのみ): 加藤元樹 (KATO, Motoki) [JP/JP]. 浜田俊也 (HAMADA, Toshiya) [JP/JP]; 〒141-0001 東京都品川区北品川6丁目7番35号 ソニー株式会社内 Tokyo (JP).

(74) 代理人: 小池 晃, 外(KOIKE, Akira et al.); 〒105-0001 東京都港区虎ノ門二丁目6番4号 第11森ビル Tokyo (JP).

(81) 指定国 (国内): CN, KR, US.

添付公開書類:
— 国際調査報告書

２文字コード及び他の略語については、定期発行される各*PCT*ガゼットの巻頭に掲載されている「コードと略語のガイダンスノート」を参照。

(54) Title: INFORMATION PROCESSING APPARATUS AND METHOD, PROGRAM, AND RECORDED MEDIUM

(54) 発明の名称: 情報処理装置及び方法、プログラム、並びに記録媒体

a シンタクス	b バイト数	c 略号
ClipInfo(){		
version_number	8*4	bslbf
length	32	uimsbf
Clip_stream_type	8	bslbf
offset_SPN	32	uimsbf
TS_recording_rate	24	uimsbf
reserved	8	bslbf
record_time_and_date	4*14	bslbf
reserved	8	bslbf
duration	4*6	bslbf
reserved	7	bslbf
time_controlled_flag	1	bslbf
TS_average_rate	24	uimsbf
if (Clip_stream_type==1) // Bridge-Clip AV stream		
RSPN_arrival_time_discontinuity	32	uimsbf
else		
reserved	32	bslbf
reserved_for_system_use	144	bslbf
reserved	11	bslbf
is_format_identifier_valid	1	bslbf
is_original_network_ID_valid	1	bslbf
is_transport_stream_ID_valid	1	bslbf
is_service_ID_valid	1	bslbf
is_country_code_valid	1	bslbf
format_identifier	32	bslbf
original_network_ID	16	uimsbf
transport_stream_ID	16	uimsbf
service_ID	16	uimsbf
country_code	24	bslbf
stream_format_name	16*8	bslbf
reserved_for_fortune_use	256	bslbf
}		

a...SYNTAX
b...NUMBER OF BYTES
c...SYMBOL

(57) Abstract: An AV stream, which is an entity of content, is managed by ClipInformation, and reproduction of the AV stream is managed by PlayList. Address information RSPN_arrival_time_discontinuity in a discontinuous point in the AV stream, which is attribute information on the AV stream, information EP_map, TU_map for relating the time information in the AV stream to the address information, and time information ClipMark on a characteristic image in the AV stream are recorded in the ClipInformation, thereby enabling quick search for a predetermined mark.

[続葉有]

WO 01/82604 A1

(57) 要約:

コンテンツの実体としてのAVストリームは、ClipInformationにより管理され、AVストリームの再生は、PlayListにより管理される。AVストリームの属性情報としての、AVストリーム中の不連続点のアドレス情報RSPN_arrival_time_discontinuity、AVストリーム中の時刻情報とアドレス情報を関連づける情報EP_map,TU_map、並びに、AVストリーム中の特徴的な画像の時刻情報ClipMarkは、ClipInformationに記録される。これにより、所定のマークを迅速に検索できるようにする。

明細書

情報処理装置及び方法、プログラム、並びに記録媒体

技術分野

本発明は情報処理装置及び方法、並びにプログラムに関し、特に、AVストリーム内のＩピクチャのアドレス情報、符号化パラメータ、変化点情報、マークなどの情報をファイルとして記録する情報処理装置及び方法、記録媒体、並びにプログラムに関する。

背景技術

近年、記録再生装置から取り外し可能なディスク型の記録媒体として、各種の光ディスクが提案されつつある。このような記録可能な光ディスクは、数ギガバイトの大容量メディアとして提案されており、ビデオ信号等のAV（Audio Visual）信号を記録するメディアとしての期待が高い。この記録可能な光ディスクに記録するデジタルのAV信号のソース（供給源）としては、CSデジタル衛星放送やBSデジタル放送があり、また、将来はデジタル方式の地上波テレビジョン放送等も提案されている。

ここで、これらのソースから供給されるデジタルビデオ信号は、通常MPEG（Moving Picture Experts Group）2方式で画像圧縮されているのが一般的である。また、記録装置には、その装置固有の記録レートが定められている。従来の民生用映像蓄積メディアで、デジタル放送のデジタルビデオ信号を記録する場合、アナログ記録方式であれば、デジタルビデオ信号をデコード後、帯域制限をして記録する。あるいは、MPEG1 Video、MPEG2 Video、DV方式をはじめとするデジタル記録方式であれば、１度デコードされた後に、その装置固有の記録レート符号化方式で再エンコードされて記録される。

しかしながら、このような記録方法は、供給されたビットストリームを１度デ

コードし、その後で帯域制限や再エンコードを行って記録するため、画質の劣化を伴う。画像圧縮されたデジタル信号の記録をする場合、入力されたデジタル信号の伝送レートが記録再生装置の記録レートを超えない場合には、供給されたビットストリームをデコードや再エンコードすることなく、そのまま記録する方法が最も画質の劣化が少ない。ただし、画像圧縮されたデジタル信号の伝送レートが記録媒体としてのディスクの記録レートを超える場合には、記録再生装置でデコード後、伝送レートがディスクの記録レートの上限以下になるように、再エンコードをして記録する必要はある。

また、入力デジタル信号のビットレートが時間により増減する可変レート方式によって伝送されている場合には、回転ヘッドが固定回転数であるために記録レートが固定レートになるテープ記録方式に比べ、１度バッファにデータを蓄積し、バースト的に記録ができるディスク記録装置の方が記録媒体の容量をより無駄なく利用できる。

以上のように、デジタル放送が主流となる将来においては、データストリーマのように放送信号をデジタル信号のまま、デコードや再エンコードすることなく記録し、記録媒体としてディスクを使用した記録再生装置が求められると予測される。

上述したような装置により、複数のデータ（例えば、映像データや音声データなどから構成される番組のデータ）が記録されている記録媒体を再生する際、ユーザのランダムアクセスや特殊再生の指示に対して、記録媒体からのAVストリームの読出位置の決定やストリームの復号といった処理を速やかに行わなくてはならないが、記録媒体に記録されるデータ量が増加するに従い、そのような処理を速やかにできないといった課題があった。

発明の開示

本発明はこのような状況に鑑みてなされたものであり、AVストリーム内のＩピクチャのアドレス情報、符号化パラメータ、変化点情報、マークなどの情報をファイルとして記録することにより、AVストリームの読出位置の決定や復号処理を

速やかに行えるようにし、特に、所定のマークを迅速に検索できるようにすることを目的とする。

本発明に係る情報処理装置は、AVストリーム中の符号化情報が連続な区間の開始アドレス情報、AVストリーム中の時刻情報とアドレス情報を関連付ける情報、及びAVストリーム中の特徴的な画像の時刻情報を生成する生成手段と、生成手段により生成された情報をClip情報として記録媒体に記録する記録手段とを備える。

符号化情報が連続な区間の開始アドレス情報は、STCシーケンス又はプログラムシーケンスの開始アドレスであり、時刻情報とアドレス情報を関連付ける情報は、EP_map又はTU_mapであり、特徴的な画像の時刻情報は、ClipMarkとすることができる。

記録手段は、AVストリームの記録レートの平均値に関する情報を記録媒体にさらに記録することができる。

平均値に関する情報は、TS_average_rateとすることができる。

AVストリームはトランスポートストリームとすることができる。

AVストリーム中の符号化情報が連続な区間の開始アドレス情報は、トランスポートストリームのシステムタイムクロックが連続な区間であるSTCシーケンスの開始アドレスを含むようにすることができる。

AVストリーム中の符号化情報が連続な区間の開始アドレス情報は、トランスポートストリームのプログラム内容が一定な区間であるプログラムシーケンスの開始アドレスを含むようにすることができる。

AVストリーム中の符号化情報が連続な区間の開始アドレス情報は、トランスポートストリームのトランスポートパケットの到着時間に基づくアライバルタイムが連続な区間の開始アドレスを含むようにすることができる。

AVストリーム中の時刻情報とアドレス情報を関連付ける情報は、トランスポートストリームのIピクチャのアドレスとそのプレゼンテーションタイムスタンプを含むようにすることができる。

AVストリーム中の時刻情報とアドレス情報を関連付ける情報の少なくとも一部を圧縮する圧縮手段をさらに備え、記録手段は、圧縮手段により圧縮された情報を記録することができる。

AVストリーム中の時刻情報とアドレス情報を関連付ける情報は、トランスポートパケットの到着時刻に基づいたアライバルタイムと、それに対応するトランスポートパケットのAVストリームデータ中のアドレスを含むようにすることができる。

本発明に係る情報処理方法は、AVストリーム中の符号化情報が連続な区間の開始アドレス情報、AVストリーム中の時刻情報とアドレス情報を関連付ける情報、及びAVストリーム中の特徴的な画像の時刻情報を生成する生成ステップと、生成ステップにより生成された情報をClip情報として記録媒体に記録する記録ステップとを含む。

本発明に係る記録媒体のプログラムは、AVストリーム中の符号化情報が連続な区間の開始アドレス情報、AVストリーム中の時刻情報とアドレス情報を関連付ける情報、及びAVストリーム中の特徴的な画像の時刻情報を生成する生成ステップと、生成ステップにより生成された情報をClip情報として記録媒体に記録する記録ステップとを含む。

本発明に係るプログラムは、AVストリーム中の符号化情報が連続な区間の開始アドレス情報、AVストリーム中の時刻情報とアドレス情報を関連付ける情報、及びAVストリーム中の特徴的な画像の時刻情報を生成する生成ステップと、生成ステップにより生成された情報をClip情報として記録媒体に記録する記録ステップとを実行させる。

本発明に係る情報処理装置は、Clip情報として、AVストリーム中の符号化情報が連続な区間の開始アドレス情報、AVストリーム中の時刻情報とアドレス情報を関連付ける情報、及びAVストリーム中の特徴的な画像の時刻情報を再生する再生手段と、再生手段により再生されたClip情報に基づいて、AVストリームの出力を制御する制御手段とを備える。

本発明に係る情報処理方法は、Clip情報として、AVストリーム中の符号化情報が連続な区間の開始アドレス情報、AVストリーム中の時刻情報とアドレス情報を関連付ける情報、及びAVストリーム中の特徴的な画像の時刻情報を再生する再生ステップと、再生ステップの処理により再生されたClip情報に基づいて、AVストリームの出力を制御する制御ステップとを含む。

本発明に係る記録媒体のプログラムは、Clip情報として、AVストリーム中の符号化情報が連続な区間の開始アドレス情報、AVストリーム中の時刻情報とアドレス情報を関連付ける情報、及びAVストリーム中の特徴的な画像の時刻情報を再生する再生ステップと、再生ステップの処理により再生されたClip情報に基づいて、AVストリームの出力を制御する制御ステップとを含む。

本発明に係るプログラムは、Clip情報として、AVストリーム中の符号化情報が連続な区間の開始アドレス情報、AVストリーム中の時刻情報とアドレス情報を関連付ける情報、及びAVストリーム中の特徴的な画像の時刻情報を再生する再生ステップと、再生ステップの処理により再生されたClip情報に基づいて、AVストリームの出力を制御する制御ステップとを実行させる。

本発明に係る記録媒体は、Clip情報として、AVストリーム中の符号化情報が連続な区間の開始アドレス情報、AVストリーム中の時刻情報とアドレス情報を関連付ける情報、及びAVストリーム中の特徴的な画像の時刻情報が記録されている。

本発明では、Clip情報として、AVストリーム中の符号化情報が連続な区間の開始アドレス情報、AVストリーム中の時刻情報とアドレス情報を関連付ける情報、及びAVストリーム中の特徴的な画像の時刻情報が記録される。

また、本発明では、Clip情報として、AVストリーム中の符号化情報が連続な区間の開始アドレス情報、AVストリーム中の時刻情報とアドレス情報を関連付ける情報、及びAVストリーム中の特徴的な画像の時刻情報が再生される。

図面の簡単な説明

図１は、　本発明を適用した記録再生装置の一実施の形態の構成を示す図である。

図２は、　記録再生装置１により記録媒体に記録されるデータのフォーマットについて説明する図である。

図３は、　Real PlayListとVirtual PlayListについて説明する図である。

図４Ａ，４Ｂ，４Ｃは、　Real PlayListの作成について説明する図である。

図５Ａ，５Ｂ，５Ｃは、　Real PlayListの削除について説明する図である。

図６Ａ，６Ｂは、　アセンブル編集について説明する図である。

図７は、　Virtual PlayListにサブパスを設ける場合について説明する図である。

図８は、　PlayListの再生順序の変更について説明する図である。

図９は、　PlayList上のマークとClip上のマークについて説明する図である。

図１０は、メニューサムネイルについて説明する図である。

図１１は、PlayListに付加されるマークについて説明する図である。

図１２は、クリップに付加されるマークについて説明する図である。

図１３は、PlayList、Clip、サムネイルファイルの関係について説明する図である。

図１４は、ディレクトリ構造について説明する図である。

図１５は、info.dvrのシンタクスを示す図である。

図１６は、DVRVolumeのシンタクスを示す図である。

図１７は、ResumeVolumeのシンタクスを示す図である。

図１８は、UIAppInfoVolumeのシンタクスを示す図である。

図１９は、Charactersetvalueのテーブルを示す図である。

図２０は、TableOfPlayListのシンタクスを示す図である。

図２１は、TableOfPlayListの他のシンタクスを示す図である。

図２２は、MakersPrivateDataのシンタクスを示す図である。

図２３は、xxxxx.rplsとyyyyy.vplsのシンタクスを示す図である。

図２４Ａ，２４Ｂ，２４Ｃは、PlayListについて説明する図である。

図２５は、PlayListのシンタクスを示す図である。

図２６は、PlayList_typeのテーブルを示す図である。

図２７は、UIAppInfoPlayListのシンタクスを示す図である。

図２８Ａ，２８Ｂ，２８Ｃは、図２７に示したUIAppInfoPlayListのシンタクス内のフラグについて説明する図である。

図２９は、PlayItemについて説明する図である。

図３０は、PlayItemについて説明する図である。

図３１は、PlayItemについて説明する図である。

図３２は、PlayItemのシンタクスを示す図である。

図３３は、IN_timeについて説明する図である。

図３４は、OUT_timeについて説明する図である。

図３５は、Connection_Conditionのテーブルを示す図である。

図３６Ａ，図３６Ｂ，図３６Ｃ，図３６Ｄは、Connection_Conditionについて説明する図である。

図３７は、BridgeSequenceInfoを説明する図である。

図３８は、BridgeSequenceInfoのシンタクスを示す図である。

図３９は、SubPlayItemについて説明する図である。

図４０は、SubPlayItemのシンタクスを示す図である。

図４１は、SubPath_typeのテーブルを示す図である。

図４２は、PlayListMarkのシンタクスを示す図である。

図４３は、Mark_typeのテーブルを示す図である。

図４４は、Mark_time_stampを説明する図である。

図４５は、zzzzz.clipのシンタクスを示す図である。

図４６は、ClipInfoのシンタクスを示す図である。

図４７は、Clip_stream_typeのテーブルを示す図である。

図４８は、offset_SPNについて説明する図である。

図４９は、offset_SPNについて説明する図である。

図５０Ａ，図５０Ｂは、STC区間について説明する図である。

図５１は、STC_Infoについて説明する図である。

図５２は、STC_Infoのシンタクスを示す図である。

図５３は、ProgramInfoを説明する図である。

図５４は、ProgramInfoのシンタクスを示す図である。

図５５は、VideoCondingInfoのシンタクスを示す図である。

図５６は、Video_formatのテーブルを示す図である。

図５７は、frame_rateのテーブルを示す図である。

図５８は、display_aspect_ratioのテーブルを示す図である。

図５９は、AudioCondingInfoのシンタクスを示す図である。

図６０は、audio_codingのテーブルを示す図である。

図６１は、audio_component_typeのテーブルを示す図である。

図６２は、sampling_frequencyのテーブルを示す図である。

図６３は、CPIについて説明する図である。

図６４は、CPIについて説明する図である。

図６５は、CPIのシンタクスを示す図である。

図６６は、CPI_typeのテーブルを示す図である。

図６７は、ビデオEP_mapについて説明する図である。

図６８は、EP_mapについて説明する図である。

図６９は、EP_mapについて説明する図である。

図７０は、EP_mapのシンタクスを示す図である。

図７１は、EP_typevaluesのテーブルを示す図である。

図７２は、EP_map_for_one_stream_PIDのシンタクスを示す図である。

図７３は、TU_mapについて説明する図である。

図７４は、TU_mapのシンタクスを示す図である。

図７５は、ClipMarkのシンタクスを示す図である。

図７６は、Mark_typeのテーブルを示す図である。

図７７は、Mark_type_stampのテーブルを示す図である。

図７８は、ClipMarkのシンタクスの他の例を示す図である。

図７９は、Mark_typeのテーブルの他の例を示す図である。

図８０は、mark_entry()とrepresentative_picture_entry()の例を示す図である。

図８１は、mark_entry()とrepresentative_picture_entry()のシンタクスを示す図である。

図８２は、mark_entry()とrepresentative_picture_entry()のシンタクスの他の例を示す図である。

図８３は、RSPN_ref_EP_startとoffset_num_picturesの関係を説明する図である。

図８４は、mark_entry()とrepresentative_picture_entry()のシンタクスの他

の例を示す図である。

図８５は、ClipMarkとEP_mapの関係を説明する図である。

図８６は、menu.thmbとmark.thmbのシンタクスを示す図である。

図８７は、Thumbnailのシンタクスを示す図である。

図８８は、thumbnail_picture_formatのテーブルを示す図である。

図８９Ａ，図８９Ｂは、tn_blockについて説明する図である。

図９０は、DVRMPEG2のトランスポートストリームの構造について説明する図である。

図９１は、DVRMPEG2のトランスポートストリームのレコーダモデルを示す図である。

図９２は、DVRMPEG2のトランスポートストリームのプレーヤモデルを示す図である。

図９３は、sourcepacketのシンタクスを示す図である。

図９４は、TP_extra_headerのシンタクスを示す図である。

図９５は、copypermissionindicatorのテーブルを示す図である。

図９６は、シームレス接続について説明する図である。

図９７は、シームレス接続について説明する図である。

図９８は、シームレス接続について説明する図である

図９９は、シームレス接続について説明する。

図１００は、シームレス接続について説明する図である。

図１０１は、オーディオのオーバーラップについて説明する図である。

図１０２は、BridgeSequenceを用いたシームレス接続について説明する図である。

図１０３は、BridgeSequenceを用いないシームレス接続について説明する図である。

図１０４は、DVRSTDモデルを示す図である。

図１０５は、復号、表示のタイミングチャートを示す図である。

図１０６は、図８１のシンタクスの場合におけるマーク点で示されるシーンの頭出し再生を説明するフローチャートである。

図１０７は、図７５又は図７８のClipMarkのmark_entry()/representative_picture_entry()が図８１のシンタクスの場合における再生の動作を説明する図である。

図１０８は、EP_mapの例を示す図である。

図１０９は、ClipMarkの例を示す図である。

図１１０は、図７５又は図７８のClipMarkのmark_entry()/representative_picture_entry()が図８１のシンタクスの場合におけるCMスキップ再生処理を説明するフローチャートである。

図１１１は、図７５又は図７８のClipMarkのmark_entry()/representative_picture_entry()が図８１のシンタクスの場合におけるCMスキップ再生処理を説明するフローチャートである。

図１１２は、図７５又は図７８のClipMarkのmark_entry()/representative_picture_entry()が図８２のシンタクスの場合におけるマーク点で示されるシーンの頭出し再生を説明するフローチャートである。

図１１３は、図７５又は図７８のClipMarkのmark_entry()/representative_picture_entry()が図８２のシンタクスの場合における再生を説明する図である。

図１１４は、EP_mapの例を示す図である。

図１１５は、ClipMarkの例を示す図である。

図１１６は、図７５又は図７８のClipMarkのmark_entry()/representative_picture_entry()が図８２のシンタクスの場合におけるCMスキップ再生を説明するフローチャートである。

図１１７は、図７５又は図７８のClipMarkのmark_entry()/representative_picture_entry()が図８２のシンタクスの場合におけるCMスキップ再生を説明するフローチャートである。

図１１８は、図７５又は図７８のClipMarkのmark_entry()/representative_picture_entry()が図８４のシンタクスの場合におけるマーク点で示されるシーンの頭出し再生を説明するフローチャートである。

図１１９は、図７５又は図７８のClipMarkのmark_entry()/representative_picture_entry()が図８４のシンタクスの場合における再生を説明する図である。

図１２０は、EP_mapの例を示す図である。

図１２１は、ClipMarkの例を示す図である。

図１２２は、図７５又は図７８のClipMarkのmark_entry()/representative_picture_entry()が図８４のシンタクスの場合におけるCMスキップ再生を説明するフローチャートである。

図１２３は、図７５又は図７８のClipMarkのmark_entry()/representative_picture_entry()が図８４のシンタクスの場合におけるCMスキップ再生を説明するフローチャートである。

図１２４は、アプリケーションフォーマットを示す図である。

図１２５は、PlayList上のマークとClip上のマークを説明する図である。

図１２６は、ClipMarkのシンタクスの他の例を示す図である。

図１２７は、ClipMarkのシンタクスのさらに他の例を示す図である。

図１２８は、ClipInfo()のシンタクスの別の例を示す図である。

図１２９は、ProgramInfo()のシンタクスの別の例を示す図である。

図１３０は、StreamCodingInfo()のシンタクスを示す図である。

図１３１は、stream_coding_typeを説明する図である。

図１３２は、EP-fineとEP-coarseの関係を説明する図である。

図１３３は、PTS_EP_fineとPTS_EP_coarseのフォーマットを説明する図である。

図１３４は、RSPN_EP_fineとRSPN_EP_coarseのフォーマットを説明する図である。

図１３５は、EP-coarseのエントリとEP-fineのエントリを説明する図である。

図１３６は、EP_mapのシンタクスの別の例を示す図である。

図１３７は、EP_stream_typevaluesを説明する図である。

図１３８は、図１３６のEP_mapのEP_map_for_one_stream_PIDのシンタクスを示す図である。

図１３９は、EP_video_typeの値の意味を説明する図である。

図１４０は、ClipAVストリームファイル及びClipInformationファイルの作成処理を説明するフローチャートである。

図１４１は、STC_Infoの作成の動作例を説明するフローチャートである。

図１４２は、ProgramInfoの作成の動作例を説明するフローチャートである。

図１４３は、EP_mapの作成の動作例を説明するフローチャートである。

図１４４は、アナログAV信号をエンコードして記録する場合における、図７５又は図７８のClipMarkのmark_entry()/representative_picture_entry()が図８１に示すシンタクスであるときのClipMarkの作成方法を説明するフローチャートである。

図１４５は、デジタルインタフェースから入力されたトランスポートストリームを記録する場合における、図７５又は図７８のClipMarkのmark_entry()/representative_picture_entry()が図８１に示すシンタクスであるときのClipMarkの作成方法を説明するフローチャートである。

図１４６は、EP_mapを使う特殊再生を説明する図である。

図１４７は、EP_mapを使用したＩピクチャサーチのためのプレーヤモデルを説明する図である。

図１４８は、ミニマイズのオペレーションの例を示す図である。

図１４９は、ミニマイズのときにIN_timeの前の不要なストリームデータを消去する例を示す図である。

図１５０は、ミニマイズのときにOUT_timeの後ろの不要なストリームデータを消去する例を説明する図である。

図１５１は、媒体を説明する図である。

発明を実施するための最良の形態

以下に、本発明の実施の形態について、図面を参照して説明する。図１は、本発明を適用した記録再生装置１の内部構成例を示す図である。まず、外部から入力された信号を記録媒体に記録する動作を行う記録部２の構成について説明する。記録再生装置１は、アナログデータ又はデジタルデータを入力し、記録することができる構成とされている。

端子１１には、アナログのビデオ信号が、端子１２には、アナログのオーディオ信号が、それぞれ入力される。端子１１に入力されたビデオ信号は、解析部１

４とAVエンコーダ１５に、それぞれ出力される。端子１２に入力されたオーディオ信号は、解析部１４とAVエンコーダ１５に出力される。解析部１４は、入力されたビデオ信号とオーディオ信号からシーンチェンジなどの特徴点を抽出する。

AVエンコーダ１５は、入力されたビデオ信号とオーディオ信号を、それぞれ符号化し、符号化ビデオストリーム(V)、符号化オーディオストリーム(A)、及びAV同期等のシステム情報(S)をマルチプレクサ１６に出力する。

符号化ビデオストリームは、例えば、MPEG (Moving Picture Expert Group) 2方式により符号化されたビデオストリームであり、符号化オーディオストリームは、例えば、MPEG1方式により符号化されたオーディオストリームや、ドルビーAC3方式（商標）により符号化されたオーディオストリーム等である。マルチプレクサ１６は、入力されたビデオ及びオーディオのストリームを、入力システム情報に基づいて多重化して、スイッチ１７を介して多重化ストリーム解析部１８とソースパケッタイザ１９に出力する。

多重化ストリームは、例えば、MPEG-2トランスポートストリームやMPEG2プログラムストリームである。ソースパケッタイザ１９は、入力された多重化ストリームを、そのストリームを記録させる記録媒体１００のアプリケーションフォーマットに従って、ソースパケットから構成されるAVストリームに符号化する。AVストリームは、ECC（誤り訂正）符号化部２０と変調部２１でECC符号の付加と変調処理が施され、書込部２２に出力される。書込部２２は、制御部２３から出力される制御信号に基づいて、記録媒体１００にAVストリームファイルを書き込む（記録する）。

デジタルインタフェース又はデジタルテレビジョンチューナから入力されるデジタルテレビジョン放送等のトランスポートストリームは、端子１３に入力される。端子１３に入力されたトランスポートストリームの記録方式には、２通りあり、それらは、トランスペアレントに記録する方式と、記録ビットレートを下げるなどの目的のために再エンコードをした後に記録する方式である。記録方式の指示情報は、ユーザインタフェースとしての端子２４から制御部２３へ入力される。

入力トランスポートストリームをトランスペアレントに記録する場合、端子１

３に入力されたトランスポートストリームは、スイッチ１７を介して多重化ストリーム解析部１８と、ソースパケッタイザ１９に出力される。これ以降の記録媒体１００へAVストリームが記録されるまでの処理は、上述のアナログの入力オーディオ信号とビデオ信号を符号化して記録する場合と同一の処理なので、その説明は省略する。

入力トランスポートストリームを再エンコードした後に記録する場合、端子１３に入力されたトランスポートストリームは、デマルチプレクサ２６に入力される。デマルチプレクサ２６は、入力されたトランスポートストリームに対してデマルチプレクス処理を施し、ビデオストリーム(V)、オーディオストリーム(A)及びシステム情報(S)を抽出する。

デマルチプレクサ２６により抽出されたストリーム（情報）のうち、ビデオストリームはAVデコーダ２７に、オーディオストリームとシステム情報はマルチプレクサ１６に、それぞれ出力される。AVデコーダ２７は、入力されたビデオストリームを復号し、その再生ビデオ信号をAVエンコーダ１５に出力する。AVエンコーダ１５は、入力ビデオ信号を符号化し、符号化ビデオストリーム(V)をマルチプレクサ１６に出力する。

一方、デマルチプレクサ２６から出力され、マルチプレクサ１６に入力されたオーディオストリームとシステム情報、及びAVエンコーダ１５から出力されたビデオストリームは、入力システム情報に基づいて、多重化されて、多重化ストリームとして多重化ストリーム解析部１８とソースパケットタイザ１９にスイッチ１７を介して出力される。これ以後の記録媒体１００へAVストリームが記録されるまでの処理は、上述のアナログの入力オーディオ信号とビデオ信号を符号化して記録する場合と同一の処理なので、その説明は省略する。

本実施の形態の記録再生装置１は、AVストリームのファイルを記録媒体１００に記録するとともに、そのファイルを説明するアプリケーションデータベース情報も記録する。アプリケーションデータベース情報は、制御部２３により作成される。制御部２３への入力情報は、解析部１４からの動画像の特徴情報、多重化ストリーム解析部１８からのAVストリームの特徴情報、及び端子２４から入力されるユーザからの指示情報である。

解析部１４から供給される動画像の特徴情報は、AVエンコーダ１５がビデオ信号を符号化する場合において、解析部１４により生成されるものである。解析部１４は、入力ビデオ信号とオーディオ信号の内容を解析し、入力動画像信号の中の特徴的な画像（クリップマーク）に関係する情報を生成する。これは、例えば、入力ビデオ信号の中のプログラムの開始点、シーンチェンジ点やCMコマーシャルのスタート点、エンド点、タイトルやテロップなどの特徴的なクリップマーク点の画像の指示情報であり、また、それにはその画像のサムネイルも含まれる。さらにオーディオ信号のステレオとモノラルの切換点や、無音区間などの情報も含まれる。

これらの画像の指示情報は、制御部２３を介して、マルチプレクサ１６へ入力される。マルチプレクサ１６は、制御部２３からクリップマークとして指定される符号化ピクチャを多重化するときに、その符号化ピクチャをAVストリーム上で特定するための情報を制御部２３に返す。具体的には、この情報は、ピクチャのＰＴＳ（プレゼンテーションタイムスタンプ）又はその符号化ピクチャのAVストリーム上でのアドレス情報である。制御部２３は、特徴的な画像の種類とその符号化ピクチャをAVストリーム上で特定するための情報を関連付けて記憶する。

多重化ストリーム解析部１８からのAVストリームの特徴情報は、記録されるAVストリームの符号化情報に関係する情報であり、解析部１８により生成される。例えば、AVストリーム内のIピクチャのタイムスタンプとアドレス情報、システムタイムクロックの不連続点情報、AVストリームの符号化パラメータ、AVストリームの中の符号化パラメータの変化点情報などが含まれる。また、端子１３から入力されるトランスポートストリームをトランスペアレントに記録する場合、多重化ストリーム解析部１８は、入力トランスポートストリームの中から前出のクリップマークの画像を検出し、その種類とクリップマークで指定するピクチャを特定するための情報を生成する。

端子２４からのユーザの指示情報は、AVストリームの中の、ユーザが指定した再生区間の指定情報、その再生区間の内容を説明するキャラクタ文字、ユーザが好みのシーンにセットするブックマークやリジューム点の情報などである。

制御部２３は、上述の入力情報に基づいて、AVストリームのデータベース（Cl

ip）、AVストリームの再生区間（PlayItem）をグループ化したもの（PlayList）のデータベース、記録媒体１００の記録内容の管理情報（info.dvr）、及びサムネイル画像の情報を作成する。これらの情報から構成されるアプリケーションデータベース情報は、AVストリームと同様にして、ECC符号化部２０、変調部２１で処理されて、書込部２２へ入力される。書込部２２は、制御部２３から出力される制御信号に基づいて、記録媒体１００へデータベースファイルを記録する。

上述したアプリケーションデータベース情報についての詳細は後述する。

このようにして記録媒体１００に記録されたAVストリームファイル（画像データと音声データのファイル）と、アプリケーションデータベース情報が再生部３により再生される場合、まず、制御部２３は、読出部２８に対して、記録媒体１００からアプリケーションデータベース情報を読み出すように指示する。そして、読出部２８は、記録媒体１００からアプリケーションデータベース情報を読み出し、そのアプリケーションデータベース情報は、復調部２９とECC復号部３０の復調と誤り訂正処理を経て、制御部２３へ入力される。

制御部２３は、アプリケーションデータベース情報に基づいて、記録媒体１００に記録されているPlayListの一覧を端子２４のユーザインタフェースへ出力する。ユーザは、PlayListの一覧から再生したいPlayListを選択し、再生を指定されたPlayListに関する情報が制御部２３へ入力される。制御部２３は、そのPlayListの再生に必要なAVストリームファイルの読出を、読出部２８に指示する。読出部２８は、その指示に従い、記録媒体１００から対応するAVストリームを読み出し復調部２９に出力する。復調部２９に入力されたAVストリームは、所定の処理が施されることにより復調され、さらにECC復号部３０の処理を経て、ソースデパケッタイザ３１出力される。

ソースデパケッタイザ３１は、記録媒体１００から読み出され、所定の処理が施されたアプリケーションフォーマットのAVストリームを、デマルチプレクサ２６が処理可能なストリームに変換する。デマルチプレクサ２６は、制御部２３により指定されたAVストリームの再生区間（PlayItem）を構成するビデオストリーム(V)、オーディオストリーム(A)及びAV同期等のシステム情報(S)を、AVデコーダ２７に出力する。AVデコーダ２７は、ビデオストリームとオーディオストリーム

を復号し、再生ビデオ信号と再生オーディオ信号を、それぞれ対応する端子３２と端子３３から出力する。

また、ユーザインタフェースとしての端子２４から、ランダムアクセス再生や特殊再生を指示する情報が入力された場合、制御部２３は、AVストリームのデータベース（Clip）の内容に基づいて、記憶媒体１００からのAVストリームの読出位置を決定し、そのAVストリームの読出を、読出部２８に指示する。例えば、ユーザにより選択されたPlayListを、所定の時刻から再生する場合、制御部２３は、指定された時刻に最も近いタイムスタンプを持つIピクチャからのデータを読み出すように読出部２８に指示する。

また、Clip Informationの中のClipMarkにストアされている番組の頭出し点やシーンチェンジ点の中から、ユーザがあるクリップマークを選択したとき（例えば、この動作は、ClipMarkにストアされている番組の頭出し点やシーンチェンジ点のサムネイル画像リストをユーザインタフェースに表示して、ユーザが、その中からある画像を選択することにより行われる）、制御部２３は、Clip Informationの内容に基づいて、記録媒体１００からのAVストリームの読出位置を決定し、そのAVストリームの読出を読出部２８へ指示する。すなわち、ユーザが選択した画像がストアされているAVストリーム上でのアドレスに最も近いアドレスにあるIピクチャからのデータを読み出すように読出部２８へ指示する。読出部２８は、指定されたアドレスからデータを読み出し、読み出されたデータは、復調部２９、ECC復号部３０、ソースデパケッタイザ３１の処理を経て、デマルチプレクサ２６へ入力され、AVデコーダ２７で復号されて、マーク点のピクチャのアドレスで示されるAVデータが再生される。

また、ユーザによって高速再生（Fast-forward playback）が指示された場合、制御部２３は、AVストリームのデータベース（Clip）に基づいて、AVストリームの中のI-ピクチャデータを順次連続して読み出すように読出部２８に指示する。

読出部２８は、指定されたランダムアクセスポイントからAVストリームのデータを読み出し、読み出されたデータは、後段の各部の処理を経て再生される。

次に、ユーザが、記録媒体１００に記録されているAVストリームの編集をする場合を説明する。ユーザが、記録媒体１００に記録されているAVストリームの再

生区間を指定して新しい再生経路を作成したい場合、例えば、番組Ａという歌番組から歌手Ａの部分を再生し、その後続けて、番組Ｂという歌番組の歌手Ａの部分を再生したいといった再生経路を作成したい場合、ユーザインタフェースとしての端子２４から再生区間の開始点（イン点）と終了点（アウト点）の情報が制御部２３に入力される。制御部２３は、AVストリームの再生区間（PlayItem）をグループ化したもの（PlayList）のデータベースを作成する。

ユーザが、記録媒体１００に記録されているAVストリームの一部を消去したい場合、ユーザインタフェースとしての端子２４から消去区間のイン点とアウト点の情報が制御部２３に入力される。制御部２３は、必要なAVストリーム部分だけを参照するようにPlayListのデータベースを変更する。また、AVストリームの不必要なストリーム部分を消去するように、書込部２２に指示する。

ユーザが、記録媒体１００に記録されているAVストリームの再生区間を指定して新しい再生経路を作成したい場合であり、かつ、それぞれの再生区間をシームレスに接続したい場合について説明する。このような場合、制御部２３は、AVストリームの再生区間（PlayItem）をグループ化したもの（PlayList）のデータベースを作成し、さらに、再生区間の接続点付近のビデオストリームの部分的な再エンコードと再多重化を行う。

まず、端子２４から再生区間のイン点のピクチャの情報と、アウト点のピクチャの情報が制御部２３へ入力される。制御部２３は、読出部２８にイン点側ピクチャとアウト点側のピクチャを再生するために必要なデータの読出を指示する。そして、読出部２８は、記録媒体１００からデータを読み出し、そのデータは、復調部２９、ECC復号部３０、ソースデパケッタイザ３１を経て、デマルチプレクサ２６に出力される。

制御部２３は、デマルチプレクサ２６に入力されたデータを解析して、ビデオストリームの再エンコード方法（picture_coding_typeの変更、再エンコードする符号化ビット量の割り当て）と、再多重化方式を決定し、その方式をAVエンコーダ１５とマルチプレクサ１６に供給する。

次に、デマルチプレクサ２６は、入力されたストリームをビデオストリーム(V)、オーディオストリーム(A)及びシステム情報(S)に分離する。ビデオストリー

ムは、AVデコーダ２７に入力されるデータとマルチプレクサ１６に入力されるデータがある。前者のデータは、再エンコードするために必要なデータであり、これはAVデコーダ２７で復号され、復号されたピクチャはAVエンコーダ１５で再エンコードされて、ビデオストリームにされる。後者のデータは、再エンコードをしないで、オリジナルのストリームからコピーされるデータである。オーディオストリーム、システム情報については、直接、マルチプレクサ１６に入力される。

マルチプレクサ１６は、制御部２３から入力された情報に基づいて、入力ストリームを多重化し、多重化ストリームを出力する。多重化ストリームは、ECC符号化部２０、変調部２１で処理されて、書込部２２に入力される。書込部２２は、制御部２３から供給される制御信号に基づいて、記録媒体１００にAVストリームを記録する。

以下に、アプリケーションデータベース情報や、その情報に基づく再生、編集といった操作に関する説明をする。図２は、アプリケーションフォーマットの構造を説明する図である。アプリケーションフォーマットは、AVストリームの管理のためにPlayListとClipの２つのレイヤをもつ。Volume Informationは、ディスク内の全てのClipとPlayListの管理をする。ここでは、１つのAVストリームとその付属情報のペアを１つのオブジェクトと考え、それをClipと称する。AVストリームファイルはClip AV stream fileと称し、その付属情報は、Clip Information fileと称する。

１つのClip AV stream fileは、MPEG-2トランスポートストリームをアプリケーションフォーマットによって規定される構造に配置したデータをストアする。一般的に、ファイルは、バイト列として扱われるが、Clip AV stream fileのコンテンツは、時間軸上に展開され、Clipの中のエントリポイント（Iピクチャ）は、主に時間ベースで指定される。所定のClipへのアクセスポイントのタイムスタンプが与えられたとき、Clip Information fileは、Clip AV stream fileの中でデータの読出を開始すべきアドレス情報を見つけるために役立つ。

PlayListについて、図３を参照して説明する。PlayListは、Clipの中からユーザが見たい再生区間を選択し、それを簡単に編集することができるようにするために設けられている。１つのPlayListは、Clipの中の再生区間の集まりである。

所定のClipの中の１つの再生区間は、PlayItemと呼ばれ、それは、時間軸上のイン点（IN）とアウト点（OUT）の対で表される。したがって、PlayListは、複数のPlayItemが集まることにより構成される。

PlayListには、２つのタイプがある。１つは、Real PlayListであり、もう１つは、Virtual PlayListである。Real PlayListは、それが参照しているClipのストリーム部分を共有している。すなわち、Real PlayListは、それの参照しているClipのストリーム部分に相当するデータ容量をディスクの中で占め、Real PlayListが消去された場合、それが参照しているClipのストリーム部分もまたデータが消去される。

Virtual PlayListは、Clipのデータを共有していない。したがって、Virtual PlayListが変更又は消去されたとしても、Clipの内容には何も変化が生じない。

次に、Real PlayListの編集について説明する。図４Ａは、Real PlayListのクリエイト（create：作成）に関する図であり、AVストリームが新しいClipとして記録される場合、そのClip全体を参照するReal PlayListが新たに作成される操作である。

図４Ｂは、Real PlayListのディバイド（divide：分割）に関する図であり、Real PlayListが所望な点で分けられて、２つのReal PlayListに分割される操作である。この分割という操作は、例えば、１つのPlayListにより管理される１つのクリップ内に、２つの番組が管理されているような場合に、ユーザが１つ１つの番組として登録（記録）し直したいといったようなときに行われる。この操作により、Clipの内容が変更される（Clip自体が分割される）ことはない。

図４Ｃは、Real PlayListのコンバイン（combine：結合）に関する図であり、２つのReal PlayListを結合して、１つの新しいReal PlayListにする操作である。この結合という操作は、例えば、ユーザが２つの番組を１つの番組として登録し直したいといったようなときに行われる。この操作により、Clipが変更される（Clip自体が１つにされる）ことはない。

図５Ａは、Real PlayList全体のデリート（delete：削除）に関する図であり、所定のReal PlayList全体を消去する操作がされた場合、削除されたReal PlayListが参照するClipの、対応するストリーム部分も削除される。

図５Ｂは、Real PlayListの部分的な削除に関する図であり、Real PlayListの所望な部分が削除された場合、対応するPlayItemが、必要なClipのストリーム部分だけを参照するように変更される。そして、Clipの対応するストリーム部分は削除される。

図５Ｃは、Real PlayListのミニマイズ（Minimize：最小化）に関する図であり、Real PlayListに対応するPlayItemを、Virtual PlayListに必要なClipのストリーム部分だけを参照するようにする操作である。Virtual PlayListにとって不必要なClipの、対応するストリーム部分は削除される。

上述したような操作により、Real PlayListが変更されて、そのReal PlayListが参照するClipのストリーム部分が削除された場合、その削除されたClipを使用しているVirtual PlayListが存在し、そのVirtual PlayListにおいて、削除されたClipにより問題が生じる可能性がある。

そのようなことが生じないように、ユーザに、削除という操作に対して、「そのReal PlayListが参照しているClipのストリーム部分を参照しているVirtual PlayListが存在し、もし、そのReal PlayListが消去されると、そのVirtual PlayListもまた消去されることになるが、それでも良いか？」といったメッセージなどを表示させることにより、確認（警告）を促した後に、ユーザの指示により削除の処理を実行又はキャンセルする。あるいはVirtual PlayListを削除する代わりに、Real PlayListに対してミニマイズの操作が行われるようにする。

次にVirtual PlayListに対する操作について説明する。Virtual PlayListに対して操作が行われたとしても、Clipの内容が変更されることはない。図６Ａ，図６Ｂは、アセンブル（Assemble）編集（IN-OUT編集）に関する図であり、ユーザが見たいと所望した再生区間のPlayItemを作り、Virtual PlayListを作成するといった操作である。PlayItem間のシームレス接続が、アプリケーションフォーマットによりサポートされている（後述）。

図６Ａに示したように、２つのReal PlayList１，２と、それぞれのReal PlayListに対応するClip１，２が存在している場合に、ユーザがReal PlayList１内の所定の区間（In１乃至Out１までの区間：PlayItem１）を再生区間として指示し、続けて再生する区間として、Real PlayList２内の所定の区間（In２乃至Out２ま

での区間：PlayItem２）を再生区間として指示したとき、図６Ｂに示すように、PlayItem１とPlayItem２から構成される１つのVirtual PlayListが作成される。

次に、Virtual PlayListの再編集（Re-editing）について説明する。再編集には、Virtual PlayListの中のイン点やアウト点の変更、Virtual PlayListへの新しいPlayItemの挿入（insert）や追加（append）、Virtual PlayListの中のPlayItemの削除などがある。また、Virtual PlayListそのものを削除することもできる。

図７は、Virtual PlayListへのオーディオのアフレコ（Audio dubbing（post recording））に関する図であり、Virtual PlayListへのオーディオのアフレコをサブパスとして登録する操作のことである。このオーディオのアフレコは、アプリケーションフォーマットによりサポートされている。Virtual PlayListのメインパスのAVストリームに、付加的なオーディオストリームが、サブパスとして付加される。

Real PlayListとVirtual PlayListで共通の操作として、図８に示すようなPlayListの再生順序の変更（Moving）がある。この操作は、ディスク（ボリューム）の中でのPlayListの再生順序の変更であり、アプリケーションフォーマットにおいて定義されるTable Of PlayList（図２０などを参照して後述する）によってサポートされる。この操作により、Clipの内容が変更されるようなことはない。

次に、マーク（Mark）について説明する。マークは、図９に示されるように、Clip及びPlayListの中のハイライトや特徴的な時間を指定するために設けられている。Clipに付加されるマークは、ClipMark（クリップマーク）と呼ばれる。ClipMarkは、AVストリームの内容に起因する特徴的なシーンを指定する、例えば番組の頭出し点やシーンチェンジ点などである。ClipMarkは、図１の例えば解析部１４によって生成される。PlayListを再生するとき、そのPlayListが参照するClipのマークを参照して、使用することができる。

PlayListに付加されるマークは、PlayListMark（プレイリストマーク）と呼ばれる。PlayListMarkは、主にユーザによってセットされる、例えば、ブックマークやリジューム点などである。Clip又はPlayListにマークをセットすることは、マークの時刻を示すタイムスタンプをマークリストに追加することにより行われ

る。また、マークを削除することは、マークリストの中から、そのマークのタイムスタンプを除去することである。したがって、マークの設定や削除により、AVストリームは何の変更もされない。

ClipMarkの別のフォーマットとして、ClipMarkが参照するピクチャをAVストリームの中でのアドレスベースで指定するようにしてもよい。Clipにマークをセットすることは、マーク点のピクチャを示すアドレスベースの情報をマークリストに追加することにより行われる。また、マークを削除することは、マークリストの中から、そのマーク点のピクチャを示すアドレスベースの情報を除去することである。したがって、マークの設定や削除により、AVストリームは何の変更もされない。

次にサムネイルについて説明する。サムネイルは、Volume、PlayList及びClipに付加される静止画である。サムネイルには、２つの種類があり、１つは、内容を表す代表画としてのサムネイルである。これは主としてユーザがカーソル（不図示）などを操作して見たいものを選択するためのメニュー画面で使われるものである。もう１つは、マークが指しているシーンを表す画像である。

Volumeと各PlayListは代表画を持つことができるようにする必要がある。Volumeの代表画は、ディスク（記録媒体１００、以下、記録媒体１００はディスク状のものであるとし、適宜、ディスクと記述する）を記録再生装置１の所定の場所にセットしたときに、そのディスクの内容を表す静止画を最初に表示する場合などに用いられることを想定している。PlayListの代表画は、PlayListを選択するメニュー画面において、PlayListの内容を表すための静止画として用いられることを想定している。

PlayListの代表画として、PlayListの最初の画像をサムネイル（代表画）にすることが考えられるが、必ずしも再生時刻０の先頭の画像が内容を表す上で最適な画像とは限らない。そこで、PlayListのサムネイルとして、任意の画像をユーザが設定できるようにする。以上Volumeを表す代表画としてのサムネイルと、PlayListを表す代表画としてのサムネイルの２種類のサムネイルをメニューサムネイルと称する。メニューサムネイルは頻繁に表示されるため、ディスクから高速に読み出される必要がある。このため、全てのメニューサムネイルを１つのファ

イルに格納することが効率的である。メニューサムネイルは、必ずしもボリューム内の動画から抜き出したピクチャである必要はなく、図１０に示すように、パーソナルコンピュータやデジタルスチルカメラから取り込まれた画像でもよい。

一方、ClipとPlayListには、複数個のマークを打てる必要があり、マーク位置の内容を知るためにマーク点の画像を容易に見ることができるようにする必要がある。このようなマーク点を表すピクチャをマークサムネイル（Mark Thumbnails）と称する。したがって、マークサムネイルの元となる画像は、外部から取り込んだ画像よりも、マーク点の画像を抜き出したものが主となる。

図１１は、PlayListに付けられるマークと、そのマークサムネイルの関係について示す図であり、図１２は、Clipに付けられるマークと、そのマークサムネイルの関係について示す図である。マークサムネイルは、メニューサムネイルと異なり、PlayListの詳細を表すときに、サブメニュー等で使われるため、短いアクセス時間で読み出されるようなことは要求されない。そのため、サムネイルが必要になる度に、記録再生装置１がファイルを開き、そのファイルの一部を読み出すことで多少時間がかかっても、問題にはならない。

また、ボリューム内に存在するファイル数を減らすために、全てのマークサムネイルは１つのファイルに格納するのがよい。PlayListはメニューサムネイル１つと複数のマークサムネイルを有することができるが、Clipは直接ユーザが選択する必要性がない（通常、PlayList経由で指定する）ため、メニューサムネイルを設ける必要はない。

図１３は、上述したことを考慮した場合のメニューサムネイル、マークサムネイル、PlayList及びClipの関係について示した図である。メニューサムネイルファイルには、PlayList毎に設けられたメニューサムネイルがファイルされている。メニューサムネイルファイルには、ディスクに記録されているデータの内容を代表するボリュームサムネイルが含まれている。マークサムネイルファイルは、各PlayList毎と各Clip毎に作成されたサムネイルがファイルされている。

次に、CPI（Characteristic Point Information）について説明する。CPIは、Clipインフォメーションファイルに含まれるデータであり、主に、それはClipへのアクセスポイントのタイムスタンプが与えられたとき、Clip AV stream fileの

中でデータの読出を開始すべきデータアドレスを見つけるために用いられる。本実施の形態では、２種類のCPIを用いる。１つは、EP_mapであり、もう１つは、TU_mapである。

EP_mapは、エントリポイント(EP)データのリストであり、それはエレメンタリストリーム及びトランスポートストリームから抽出されたものである。これは、AVストリームの中でデコードを開始すべきエントリポイントの場所を見つけるためのアドレス情報を持つ。１つのEPデータは、プレゼンテーションタイムスタンプ（PTS）と、そのPTSに対応するアクセスユニットのAVストリームの中のデータアドレスの対で構成される。

EP_mapは、主に２つの目的のために使用される。第１に、PlayListの中でプレゼンテーションタイムスタンプによって参照されるアクセスユニットのAVストリームの中のデータアドレスを見つけるために使用される。第２に、ファーストフォワード再生やファーストリバース再生のために使用される。記録再生装置１が、入力AVストリームを記録する場合、そのストリームのシンタクスを解析することができるとき、EP_mapが作成され、ディスクに記録される。

TU_mapは、デジタルインタフェースを通して入力されるトランスポートパケットの到着時刻に基づいたタイムユニット（TU）データのリストを持つ。これは、到着時刻ベースの時間とAVストリームの中のデータアドレスとの関係を与える。記録再生装置１が、入力AVストリームを記録する場合、そのストリームのシンタクスを解析することができないとき、TU_mapが作成され、ディスクに記録される。

STCInfoは、MPEG-2トランスポートストリームをストアしているAVストリームファイルの中にあるSTCの不連続点情報をストアする。

AVストリームがSTCの不連続点を持つ場合、そのAVストリームファイルの中で同じ値のPTSが現れるかもしれない。そのため、AVストリーム上のある時刻をPTSベースで指す場合、アクセスポイントのPTSだけではそのポイントを特定するためには不十分である。さらに、そのPTSを含むところの連続なSTC区間のインデックスが必要である。連続なSTC区間を、このフォーマットではSTC-sequenceと呼び、そのインデックスをSTC-sequence-idと呼ぶ。STC-sequenceの情報は、Clip Information fileのSTCInfoで定義される。

STC-sequence-idは、EP_mapを持つAVストリームファイルで使用するものであり、TU_mapを持つAVストリームファイルではオプションである。

プログラムは、エレメンタリストリームの集まりであり、これらのストリームの同期再生のために、ただ1つのシステムタイムベースを共有するものである。

再生装置（図1の記録再生装置1）にとって、AVストリームのデコードに先だち、そのAVストリームの内容がわかることは有用である。例えば、ビデオやオーディオのエレメンタリストリームを伝送するトランスポートパケットのPIDの値や、ビデオやオーディオのコンポーネント種類（例えば、HDTVのビデオとMPEG-2 AACのオーディオストリームなど）などの情報である。この情報はAVストリームを参照するところのPlayListの内容をユーザに説明するメニュー画面を作成するのに有用であるし、また、AVストリームのデコードに先だって、再生装置のAVデコーダ及びデマルチプレクサの初期状態をセットするために役立つ。

この理由のために、Clip Information fileは、プログラムの内容を説明するためのProgramInfoを持つ。

MPEG-2トランスポートストリームをストアしているAVストリームファイルは、ファイルの中でプログラム内容が変化するかもしれない。例えば、ビデオエレメンタリストリームを伝送するところのトランスポートパケットのPIDが変化したり、ビデオストリームのコンポーネント種類がSDTVからHDTVに変化するなどである。

ProgramInfoは、AVストリームファイルの中でのプログラム内容の変化点の情報をストアする。AVストリームファイルの中で、このフォーマットで定めるところのプログラム内容が一定である区間をprogram-sequenceと呼ぶ。

program-sequenceは、EP_mapを持つAVストリームファイルで使用するものであり、TU_mapを持つAVストリームファイルではオプションである。

本実施の形態では、セルフエンコードのストリームフォーマット（SESF）を定義する。SESFは、アナログ入力信号を符号化する目的、及びデジタル入力信号（例えばDV）をデコードしてからMPEG-2トランスポートストリームに符号化する場合に用いられる。

SESFは、MPEG-2トランスポートストリーム及びAVストリームについてのエレメンタリストリームの符号化制限を定義する。記録再生装置1が、SESFストリーム

をエンコードし、記録する場合、EP_mapが作成され、ディスクに記録される。

デジタル放送のストリームは、次に示す方式のうちのいずれかが用いられて記録媒体１００に記録される。まず、デジタル放送のストリームをSESFストリームにトランスコーディングする。この場合、記録されたストリームは、SESFに準拠しなければならない。この場合、EP_mapが作成されて、ディスクに記録されなければならない。

あるいは、デジタル放送ストリームを構成するエレメンタリストリームを新しいエレメンタリストリームにトランスコーディングし、そのデジタル放送ストリームの規格化組織が定めるストリームフォーマットに準拠した新しいトランスポートストリームに再多重化する。この場合、EP_mapが作成されて、ディスクに記録されなければならない。

例えば、入力ストリームがISDB（日本のデジタルBS放送の規格名称）準拠のMPEG-2トランスポートストリームであり、それがHDTVビデオストリームとMPEG AACオーディオストリームを含むとする。HDTVビデオストリームをSDTVビデオストリームにトランスコーディングし、そのSDTVビデオストリームとオリジナルのAACオーディオストリームをTSに再多重化する。SDTVストリームと記録されるトランスポートストリームは、共にISDBフォーマットに準拠しなければならない。

デジタル放送のストリームが、記録媒体１００に記録される際の他の方式として、入力トランスポートストリームをトランスペアレントに記録する（入力トランスポートストリームを何も変更しないで記録する）場合であり、そのときにEP_mapが作成されてディスクに記録される。

あるいは入力トランスポートストリームをトランスペアレントに記録する（入力トランスポートストリームを何も変更しないで記録する）場合であり、そのときにTU_mapが作成されてディスクに記録される。

次にディレクトリとファイルについて説明する。以下、記録再生装置１をDVR（Digital Video Recording）と適宜記述する。図１４はディスク上のディレクトリ構造の一例を示す図である。DVRのディスク上に必要なディレクトリは、図１４に示したように、”DVR”ディレクトリを含むrootディレクトリ、”PLAYLIST”ディレクトリ、”CLIPINF”ディレクトリ、”M2TS”ディレクトリ、及び”DATA”ディレクトリ

を含む"DVR"ディレクトリである。rootディレクトリの下に、これら以外のディレクトリを作成されるようにしてもよいが、それらは、本実施の形態のアプリケーションフォーマットでは、無視されるとする。

"DVR"ディレクトリの下には、DVRアプリケーションフォーマットによって規定される全てのファイルとディレクトリがストアされる。"DVR"ディレクトリは、4個のディレクトリを含む。"PLAYLIST"ディレクトリの下には、Real PlayListとVirtual PlayListのデータベースファイルが置かれる。このディレクトリは、PlayListが1つもなくても存在する。

"CLIPINF"ディレクトリの下には、Clipのデータベースが置かれる。このディレクトリも、Clipが1つもなくても存在する。"M2TS"ディレクトリの下には、AVストリームファイルが置かれる。このディレクトリは、AVストリームファイルが1つもなくても存在する。"DATA"ディレクトリは、デジタルＴＶ放送などのデータ放送のファイルがストアされる。

"DVR"ディレクトリは、次に示すファイルをストアする。"info.dvr"ファイルは、DVRディレクトリの下に作られ、アプリケーションレイヤの全体的な情報をストアする。DVRディレクトリの下には、ただ1つのinfo.dvrがなければならない。ファイル名は、info.dvrに固定されるとする。"menu.thmb"ファイルは、メニューサムネイル画像に関連する情報をストアする。DVRディレクトリの下には、0又は1つのメニューサムネイルがなければならない。ファイル名は、memu.thmbに固定されるとする。メニューサムネイル画像が1つもない場合、このファイルは、存在しなくてもよい。

"mark.thmb"ファイルは、マークサムネイル画像に関連する情報をストアする。DVRディレクトリの下には、0又は1つのマークサムネイルがなければならない。ファイル名は、mark.thmbに固定されるとする。メニューサムネイル画像が1つもない場合、このファイルは、存在しなくてもよい。

"PLAYLIST"ディレクトリは、2種類のPlayListファイルをストアするものであり、それらは、Real PlayListとVirtual PlayListである。"xxxxx.rpls"ファイルは、1つのReal PlayListに関連する情報をストアする。それぞれのReal PlayList毎に、1つのファイルが作られる。ファイル名は、"xxxxx.rpls"である。ここ

で、"xxxxx"は、５個の０乃至９まで数字である。ファイル拡張子は、"rpls"でなければならないとする。

"yyyyy.vpls"ファイルは、１つのVirtual PlayListに関連する情報をストアする。それぞれのVirtual PlayList毎に、１つのファイルが作られる。ファイル名は、"yyyyy.vpls"である。ここで、"yyyyy"は、５個の０乃至９まで数字である。ファイル拡張子は、"vpls"でなければならないとする。

"CLIPINF"ディレクトリは、それぞれのAVストリームファイルに対応して、１つのファイルをストアする。"zzzzz.clpi"ファイルは、１つのAVストリームファイル（Clip AV stream file又はBridge-Clip AV stream file）に対応するClip Information fileである。ファイル名は、"zzzzz.clpi"であり、"zzzzz"は、５個の０乃至９までの数字である。ファイル拡張子は、"clpi"でなければならないとする。

"M2TS"ディレクトリは、AVストリームのファイルをストアする。"zzzzz.m2ts"ファイルは、DVRシステムにより扱われるAVストリームファイルである。これは、Clip AV stream file又はBridge-Clip AV streamである。ファイル名は、"zzzzz.m2ts"であり、"zzzzz"は、５個の０乃至９までの数字である。ファイル拡張子は、"m2ts"でなければならないとする。

"DATA"ディレクトリは、データ放送から伝送されるデータをストアするものであり、データとは、例えば、XML fileやMHEGファイルなどである。

次に、各ディレクトリ（ファイル）のシンタクスとセマンティクスを説明する。まず、"info.dvr"ファイルについて説明する。図１５は、"info.dvr"ファイルのシンタクスを示す図である。"info.dvr"ファイルは、３個のオブジェクトから構成され、それらは、DVRVolume()、TableOfPlayLists()及びMakersPrivateData()である。

図１５に示したinfo.dvrのシンタクスについて説明するに、TableOfPlayLists_Start_addressは、info.dvrファイルの先頭のバイトからの相対バイト数を単位として、TableOfPlayList()の先頭アドレスを示す。相対バイト数は０からカウントされる。

MakersPrivateData_Start_addressは、info.dvrファイルの先頭のバイトからの

相対バイト数を単位として、MakersPrivateData()の先頭アドレスを示す。相対バイト数は０からカウントされる。padding_word（パディングワード）は、info.dvrのシンタクスに従って挿入される。Ｎ１とＮ２は、０又は任意の正の整数である。それぞれのパディングワードは、任意の値を取るようにしてもよい。

DVRVolume()は、ボリューム（ディスク）の内容を記述する情報をストアする。図１６は、DVRVolume()のシンタクスを示す図である。図１６に示したDVR_Volume()のシンタクスを説明するに、version_numberは、このDVRVolume()のバージョンナンバを示す４個のキャラクタ文字を示す。version_numberは、ISO 646に従って、"0045"と符号化される。

lengthは、このlengthフィールドの直後からDVRVolume()の最後までのDVRVolume()のバイト数を示す３２ビットの符号なし整数で表される。

ResumeVolume()は、ボリュームの中で最後に再生したReal PlayList又はVirtual PlayListのファイル名を記憶している。ただし、Real PlayList又はVirtual PlayListの再生をユーザが中断したときの再生位置は、PlayListMark()において定義されるresume-markにストアされる（図４２、図４３）。

図１７は、ResumeVolume()のシンタクスを示す図である。図１７に示したResumeVolume()のシンタクスを説明するに、valid_flagは、この１ビットのフラグが１にセットされている場合、resume_PlayList_nameフィールドが有効であることを示し、このフラグが０にセットされている場合、resume_PlayList_nameフィールドが無効であることを示す。

resume_PlayList_nameの１０バイトのフィールドは、リジュームされるべきReal PlayList又はVirtual PlayListのファイル名を示す。

図１６に示したDVRVolume()のシンタクスの中の、UIAppInfoVolumeは、ボリュームについてのユーザインタフェースアプリケーションのパラメータをストアする。図１８は、UIAppInfoVolumeのシンタクスを示す図であり、そのセマンティクスを説明するに、character_setの８ビットのフィールドは、Volume_nameフィールドに符号化されているキャラクタ文字の符号化方法を示す。その符号化方法は、図１９に示される値に対応する。

name_lengthの８ビットフィールドは、Volume_nameフィールドの中に示される

ボリューム名のバイト長を示す。Volume_nameのフィールドは、ボリュームの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクタ文字であり、それはボリュームの名称を示す。Volume_nameフィールドの中で、それら有効なキャラクタ文字の後の値は、どんな値が入っていてもよい。

Volume_protect_flagは、ボリュームの中のコンテンツを、ユーザに制限することなしに見せてよいかどうかを示すフラグである。このフラグが１にセットされている場合、ユーザが正しくPIN番号（パスワード）を入力できたときだけ、そのボリュームのコンテンツを、ユーザに見せること（再生されること）が許可される。このフラグが０にセットされている場合、ユーザがPIN番号を入力しなくても、そのボリュームのコンテンツを、ユーザに見せることが許可される。

最初に、ユーザが、ディスクをプレーヤへ挿入した時点において、もしこのフラグが０にセットされているか、あるいはこのフラグが１にセットされていてもユーザがPIN番号を正しく入力できたならば、記録再生装置１は、そのディスクの中のPlayListの一覧を表示させる。それぞれのPlayListの再生制限は、Volume_protect_flagとは無関係であり、それはUIAppInfoPlayList()の中に定義されるplayback_control_flagによって示される。

PINは、４個の０乃至９までの数字で構成され、それぞれの数字は、ISO/IEC 646に従って符号化される。ref_thumbnail_indexのフィールドは、ボリュームに付加されるサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのボリュームにはサムネイル画像が付加されており、そのサムネイル画像は、menu.thumファイルの中にストアされている。その画像は、menu.thumファイルの中でref_thumbnail_indexの値を用いて参照される。ref_thumbnail_indexフィールドが、0xFFFFである場合、そのボリュームにはサムネイル画像が付加されていないことを示す。

次に図１５に示したinfo.dvrのシンタクス内のTableOfPlayLists()について説明する。TableOfPlayLists()は、PlayList (Real PlayListとVirtual PlayList)のファイル名をストアする。ボリュームに記録されている全てのPlayListファイルは、TableOfPlayList()の中に含まれる。TableOfPlayLists()は、ボリュームの中のPlayListのデフォルトの再生順序を示す。

図２０は、TableOfPlayLists()のシンタクスを示す図であり、そのシンタクスについて説明するに、TableOfPlayListsのversion_numberは、このTableOfPlayListsのバージョンナンバを示す４個のキャラクタ文字を示す。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。

lengthは、このlengthフィールドの直後からTableOfPlayLists()の最後までのTableOfPlayLists()のバイト数を示す３２ビットの符号なしの整数である。number_of_PlayListsの１６ビットのフィールドは、PlayList_file_nameを含むfor-loopのループ回数を示す。この数字は、ボリュームに記録されているPlayListの数に等しくなければならない。PlayList_file_nameの１０バイトの数字は、PlayListのファイル名を示す。

図２１は、TableOfPlayLists()のシンタクスの別の構成を示す図である。図２１に示したシンタクスは、図２０に示したシンタクスに、UIAppInfoPlayList（後述）を含ませた構成とされている。このように、UIAppInfoPlayListを含ませた構成とすることで、TableOfPlayListsを読み出すだけで、メニュー画面を作成することが可能となる。ここでは、図２０に示したシンタクスを用いるとして以下の説明をする。

図１５に示したinfo.dvrのシンタクス内のMakersPrivateDataについて説明する。MakersPrivateDataは、記録再生装置１のメーカが、各社の特別なアプリケーションのために、MakersPrivateData()の中にメーカのプライベートデータを挿入できるように設けられている。各メーカのプライベートデータは、それを定義したメーカを識別するために標準化されたmaker_IDを持つ。MakersPrivateData()は、１つ以上のmaker_IDを含んでもよい。

所定のメーカが、プライベートデータを挿入したいときに、既に他のメーカのプライベートデータがMakersPrivateData()に含まれていた場合、他のメーカは、既にある古いプライベートデータを消去するのではなく、新しいプライベートデータをMakersPrivateData()の中に追加するようにする。このように、本実施の形態においては、複数のメーカのプライベートデータが、１つのMakersPrivateData()に含まれることが可能であるようにする。

図２２は、MakersPrivateDataのシンタクスを示す図である。図２２に示したM

akersPrivateDataのシンタクスについて説明するに、version_numberは、このMakersPrivateData()のバージョンナンバを示す４個のキャラクタ文字を示す。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。lengthは、このlengthフィールドの直後からMakersPrivateData()の最後までのMakersPrivateData()のバイト数を示す３２ビットの符号なし整数を示す。

mpd_blocks_start_addressは、MakersPrivateData()の先頭のバイトからの相対バイト数を単位として、最初のmpd_block()の先頭バイトアドレスを示す。相対バイト数は０からカウントされる。number_of_maker_entriesは、MakersPrivateData()の中に含まれているメーカプライベートデータのエントリ数を与える１６ビットの符号なし整数である。MakersPrivateData()の中に、同じmaker_IDの値を持つメーカプライベートデータが２個以上存在してはならない。

mpd_block_sizeは、１０２４バイトを単位として、１つのmpd_blockの大きさを与える１６ビットの符号なし整数である。例えば、mpd_block_size=１ならば、それは１つのmpd_blockの大きさが１０２４バイトであることを示す。number_of_mpd_blocksは、MakersPrivateData()の中に含まれるmpd_blockの数を与える１６ビットの符号なし整数である。maker_IDは、そのメーカプライベートデータを作成したDVRシステムの製造メーカを示す１６ビットの符号なし整数である。maker_IDに符号化される値は、このDVRフォーマットのライセンサによって指定される。

maker_model_codeは、そのメーカプライベートデータを作成したDVRシステムのモデルナンバコードを示す１６ビットの符号なし整数である。maker_model_codeに符号化される値は、このフォーマットのライセンスを受けた製造メーカによって設定される。start_mpd_block_numberは、そのメーカプライベートデータが開始されるmpd_blockの番号を示す１６ビットの符号なし整数である。メーカプライベートデータの先頭データは、mpd_blockの先頭にアラインされなければならない。start_mpd_block_numberは、mpd_blockのfor-loopの中の変数jに対応する。

mpd_lengthは、バイト単位でメーカプライベートデータの大きさを示す３２ビットの符号なし整数である。mpd_blockは、メーカプライベートデータがストアされる領域である。MakersPrivateData()の中の全てのmpd_blockは、同じサイズでなければならない。

次に、Real PlayList fileとVirtual PlayList fileについて、換言すれば、xxxxx.rplsとyyyyy.vplsについて説明する。図２３は、xxxxx.rpls（Real PlayList）又はyyyyy.vpls（Virtual PlayList）のシンタクスを示す図である。xxxxx.rplsとyyyyy.vplsは、同一のシンタクス構成をもつ。xxxxx.rplsとyyyyy.vplsは、それぞれ、３個のオブジェクトから構成され、それらは、PlayList()、PlayListMark()及びMakersPrivateData()である。

PlayListMark_Start_addressは、PlayListファイルの先頭のバイトからの相対バイト数を単位として、PlayListMark()の先頭アドレスを示す。相対バイト数は０からカウントされる。

MakersPrivateData_Start_addressは、PlayListファイルの先頭のバイトからの相対バイト数を単位として、MakersPrivateData()の先頭アドレスを示す。相対バイト数は０からカウントされる。

padding_word（パディングワード）は、PlayListファイルのシンタクスに従って挿入され、Ｎ１とＮ２は、０又は任意の正の整数である。それぞれのパディングワードは、任意の値を取るようにしてもよい。

ここで、既に、簡便に説明したが、PlayListについてさらに説明する。ディスク内にある全てのReal PlayListによって、Bridge-Clip（後述）を除く全てのClipの中の再生区間が参照されていなければならない。かつ、２つ以上のReal PlayListが、それらのPlayItemで示される再生区間を同一のClipの中でオーバーラップさせてはならない。

図２４Ａ，図２４Ｂ，図２４Ｃを参照してさらに説明するに、図２４Ａに示したように、全てのClipは、対応するReal PlayListが存在する。この規則は、図２４Ｂに示したように、編集作業が行われた後においても守られる。したがって、全てのClipは、どれかしらのReal PlayListを参照することにより、必ず視聴することが可能である。

図２４Ｃに示したように、Virtual PlayListの再生区間は、Real PlayListの再生区間又はBridge-Clipの再生区間の中に含まれていなければならない。どのVirtual PlayListにも参照されないBridge-Clipがディスクの中に存在してはならない。

Real PlayListは、PlayItemのリストを含むが、SubPlayItemを含んではならない。Virtual PlayListは、PlayItemのリストを含み、PlayList()の中に示されるCPI_typeがEP_map typeであり、かつPlayList_typeが０（ビデオとオーディオを含むPlayList）である場合、Virtual PlayListは、１つのSubPlayItemを含むことができる。本実施の形態におけるPlayList()では、SubPlayIteはオーディオのアフレコの目的にだけに使用される、そして、１つのVirtual PlayListが持つSubPlayItemの数は、０又は１でなければならない。

次に、PlayListについて説明する。図２５は、PlayListのシンタクスを示す図である。図２５に示したPlayListのシンタクスを説明するに、version_numberは、このPlayList()のバージョンナンバを示す４個のキャラクタ文字である。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。lengthは、このlengthフィールドの直後からPlayList()の最後までのPlayList()のバイト数を示す３２ビットの符号なし整数である。PlayList_typeは、このPlayListのタイプを示す８ビットのフィールドであり、その一例を図２６に示す。

CPI_typeは、１ビットのフラグであり、PlayItem()及びSubPlayItem()によって参照されるClipのCPI_typeの値を示す。１つのPlayListによって参照される全てのClipは、それらのCPI()の中に定義されるCPI_typeの値が同じでなければならない。number_of_PlayItemsは、PlayListの中にあるPlayItemの数を示す１６ビットのフィールドである。

所定のPlayItem()に対応するPlayItem_idは、PlayItem()を含むfor-loopの中で、そのPlayItem()の現れる順番により定義される。PlayItem_idは、０から開始される。number_of_SubPlayItemsは、PlayListの中にあるSubPlayItemの数を示す１６ビットのフィールドである。この値は、０又は１である。付加的なオーディオストリームのパス（オーディオストリームパス）は、サブパスの一種である。

次に、図２５に示したPlayListのシンタクスのUIAppInfoPlayListについて説明する。UIAppInfoPlayListは、PlayListについてのユーザインタフェースアプリケーションのパラメータをストアする。図２７は、UIAppInfoPlayListのシンタクスを示す図である。図２７に示したUIAppInfoPlayListのシンタクスを説明するに、character_setは、８ビットのフィールドであり、PlayList_nameフィールドに符

号化されているキャラクタ文字の符号化方法を示す。その符号化方法は、図１９に示したテーブルに準拠する値に対応する。

name_lengthは、８ビットフィールドであり、PlayList_nameフィールドの中に示されるPlayList名のバイト長を示す。PlayList_nameのフィールドは、PlayListの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクタ文字であり、それはPlayListの名称を示す。PlayList_nameフィールドの中で、それら有効なキャラクタ文字の後の値は、どんな値が入っていてもよい。

record_time_and_dateは、PlayListが記録されたときの日時をストアする５６ビットのフィールドである。このフィールドは、年／月／日／時／分／秒について、１４個の数字を４ビットのBinary Coded Decimal (BCD) で符号化したものである。例えば、2001/12/23:01:02:03は、"0x20011223010203"と符号化される。

durationは、PlayListの総再生時間を時間／分／秒の単位で示した２４ビットのフィールドである。このフィールドは、６個の数字を４ビットのBinary Coded Decimal (BCD) で符号化したものである。例えば、01:45:30は、"0x014530"と符号化される。

valid_periodは、PlayListが有効である期間を示す３２ビットのフィールドである。このフィールドは、８個の数字を４ビットのBinary Coded Decimal (BCD) で符号化したものである。例えば、記録再生装置１は、この有効期間の過ぎたPlayListを自動消去する、といったように用いられる。例えば、2001/05/07は、"0x20010507"と符号化される。

maker_IDは、そのPlayListを最後に更新したDVRプレーヤ（記録再生装置１）の製造者を示す１６ビットの符号なし整数である。maker_IDに符号化される値は、DVRフォーマットのライセンサによって割り当てられる。maker_codeは、そのPlayListを最後に更新したDVRプレーヤのモデル番号を示す１６ビットの符号なし整数である。maker_codeに符号化される値は、DVRフォーマットのライセンスを受けた製造者によって決められる。

playback_control_flagのフラグが１にセットされている場合、ユーザが正しくPIN番号を入力できた場合にだけ、そのPlayListは再生される。このフラグが０に

セットされている場合、ユーザがPIN番号を入力しなくても、ユーザは、そのPlayListを視聴することができる。

write_protect_flagは、図２８Ａにテーブルを示すように、１にセットされている場合、write_protect_flagを除いて、そのPlayListの内容は、消去及び変更されない。このフラグが０にセットされている場合、ユーザは、そのPlayListを自由に消去及び変更できる。このフラグが１にセットされている場合、ユーザが、そのPlayListを消去、編集又は上書きする前に、記録再生装置１はユーザに再確認するようなメッセージを表示させる。

write_protect_flagが０にセットされているReal PlayListが存在し、かつ、そのReal PlayListのClipを参照するVirtual PlayListが存在し、そのVirtual PlayListのwrite_protect_flagが１にセットされていてもよい。ユーザが、Real PlayListを消去しようとする場合、記録再生装置１は、そのReal PlayListを消去する前に、上述のVirtual PlayListの存在をユーザに警告するか、あるいはそのReal PlayListを”Minimize”する。

is_played_flagは、図２８Ｂに示すように、フラグが１にセットされている場合、そのPlayListは、記録されてから一度は再生されたことを示し、０にセットされている場合、そのPlayListは、記録されてから一度も再生されたことがないことを示す。

archiveは、図２８Ｃに示すように、そのPlayListがオリジナルであるか、コピーされたものであるかを示す２ビットのフィールドである。ref_thumbnail_indexのフィールドは、PlayListを代表するサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのPlayListには、PlayListを代表するサムネイル画像が付加されており、そのサムネイル画像は、menu.thumファイルの中にストアされている。その画像は、menu.thumファイルの中でref_thumbnail_indexの値を用いて参照される。ref_thumbnail_indexフィールドが、0xFFFFである場合、そのPlayListには、PlayListを代表するサムネイル画像が付加されていない。

次にPlayItemについて説明する。１つのPlayItem()は、基本的に次のデータを含む。Clipのファイル名を指定するためのClip_Information_file_name、Clipの

再生区間を特定するためのIN_timeとOUT_timeのペア、PlayList()において定義されるCPI_typeがEP_map typeである場合、IN_timeとOUT_timeが参照するところのSTC_sequence_id、及び先行するPlayItemと現在のPlayItemとの接続の状態を示すところのConnection_Conditionである。

PlayListが２つ以上のPlayItemから構成されるとき、それらのPlayItemはPlayListのグローバル時間軸上に、時間のギャップ又はオーバーラップなしに一列に並べられる。PlayList()において定義されるCPI_typeがEP_map typeであり、かつ現在のPlayItemがBridgeSequence()を持たないとき、そのPlayItemにおいて定義されるIN_timeとOUT_timeのペアは、STC_sequence_idによって指定される同じSTC連続区間上の時間を指していなければならない。そのような例を図２９に示す。

図３０は、PlayList()において定義されるCPI_typeがEP_map typeであり、かつ現在のPlayItemがBridgeSequence()を持つとき、次に説明する規則が適用される場合を示している。現在のPlayItemに先行するPlayItemのIN_time（図の中でIN_time1と示されているもの）は、先行するPlayItemのSTC_sequence_idによって指定されるSTC連続区間上の時間を指している。先行するPlayItemのOUT_time（図の中でOUT_time1と示されているもの）は、現在のPlayItemのBridgeSequenceInfo()の中で指定されるBridge-Clipの中の時間を指している。このOUT_timeは、後述する符号化制限に従っていなければならない。

現在のPlayItemのIN_time（図の中でIN_time2と示されているもの）は、現在のPlayItemのBridgeSequenceInfo()の中で指定されるBridge-Clipの中の時間を指している。このIN_timeも、後述する符号化制限に従っていなければならない。現在のPlayItemのPlayItemのOUT_time（図の中でOUT_time2と示されているもの）は、現在のPlayItemのSTC_sequence_idによって指定されるSTC連続区間上の時間を指している。

図３１に示すように、PlayList()のCPI_typeがTU_map typeである場合、PlayItemのIN_timeとOUT_timeのペアは、同じClip AVストリーム上の時間を指している。

PlayItemのシンタクスは、図３２に示すようになる。図３２に示したPlayItemのシンタクスを説明するに、Clip_Information_file_nameのフィールドは、Clip Information fileのファイル名を示す。このClip Information fileのClipInfo

()において定義されるClip_stream_typeは、Clip AV streamを示していなければならない。

STC_sequence_idは、８ビットのフィールドであり、PlayItemが参照するSTC連続区間のSTC_sequence_idを示す。PlayList()の中で指定されるCPI_typeがTU_map typeである場合、この８ビットフィールドは何も意味を持たず、０にセットされる。IN_timeは、３２ビットフィールドであり、PlayItemの再生開始時刻をストアする。IN_timeのセマンティクスは、図３３に示すように、PlayList()において定義されるCPI_typeによって異なる。

OUT_timeは、３２ビットフィールドであり、PlayItemの再生終了時刻をストアする。OUT_timeのセマンティクスは、図３４に示すように、PlayList()において定義されるCPI_typeによって異なる。

Connection_Conditionは、図３５に示したような先行するPlayItemと、現在のPlayItemとの間の接続状態を示す２ビットのフィールドである。図３６Ａ，図３６Ｂ，図３６Ｃ，図３６Ｄは、図３５に示したConnection_Conditionの各状態について説明する図である。

次に、BridgeSequenceInfoについて、図３７を参照して説明する。BridgeSequenceInfo()は、現在のPlayItemの付属情報であり、次に示す情報を持つ。Bridge-Clip AV streamファイルとそれに対応するClip Information file（図４５）を指定するBridge_Clip_Information_file_nameを含む。

また、先行するPlayItemが参照するClip AV stream上のソースパケットのアドレスであり、このソースパケットに続いてBridge-Clip AV streamファイルの最初のソースパケットが接続される。このアドレスは、RSPN_exit_from_previous_Clipと称される。さらに現在のPlayItemが参照するClip AV stream上のソースパケットのアドレスであり、このソースパケットの前にBridge-Clip AV streamファイルの最後のソースパケットが接続される。このアドレスは、RSPN_enter_to_current_Clipと称される。

図３７において、RSPN_arrival_time_discontinuityは、the Bridge-Clip AV streamファイルの中でアライバルタイムベースの不連続点があるところのソースパケットのアドレスを示す。このアドレスは、ClipInfo()（図４６）の中におい

て定義される。

図３８は、BridgeSequenceInfoのシンタクスを示す図である。図３８に示したBridgeSequenceInfoのシンタクスを説明するに、Bridge_Clip_Information_file_nameのフィールドは、Bridge-Clip AV streamファイルに対応するClip Information fileのファイル名を示す。このClip Information fileのClipInfo()において定義されるClip_stream_typeは、'Bridge-Clip AV stream'を示していなければならない。

RSPN_exit_from_previous_Clipの３２ビットフィールドは、先行するPlayItemが参照するClip AV stream上のソースパケットの相対アドレスであり、このソースパケットに続いてBridge-Clip AV streamファイルの最初のソースパケットが接続される。RSPN_exit_from_previous_Clipは、ソースパケット番号を単位とする大きさであり、先行するPlayItemが参照するClip AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。

RSPN_enter_to_current_Clipの３２ビットフィールドは、現在のPlayItemが参照するClip AV stream上のソースパケットの相対アドレスであり、このソースパケットの前にBridge-Clip AV streamファイルの最後のソースパケットが接続される。RSPN_exit_from_previous_Clipは、ソースパケット番号を単位とする大きさであり、現在のPlayItemが参照するClip AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。

次に、SubPlayItemについて、図３９を参照して説明する。SubPlayItem()の使用は、PlayList()のCPI_typeがEP_map typeである場合だけに許される。本実施の形態においては、SubPlayItemはオーディオのアフレコの目的のためだけに使用されるとする。SubPlayItem()は、次に示すデータを含む。まず、PlayListの中のsub pathが参照するClipを指定するためのClip_Information_file_nameを含む。

また、Clipの中のsub pathの再生区間を指定するためのSubPath_IN_timeとSubPath_OUT_timeを含む。さらに、main pathの時間軸上でsub pathが再生開始する時刻を指定するためのsync_PlayItem_idとsync_start_PTS_of_PlayItemを含む。

sub pathに参照されるオーディオのClip AV streamは、STC不連続点（システムタイムベースの不連続点）を含んではならない。sub pathに使われるClipのオーディオサンプルのクロックは、main pathのオーディオサンプルのクロックにロックされている。

図４０は、SubPlayItemのシンタクスを示す図である。図４０に示したSubPlayItemのシンタクスを説明するに、Clip_Information_file_nameのフィールドは、Clip Information fileのファイル名を示し、それはPlayListの中でsub pathによって使用される。このClip Information fileのClipInfo()において定義されるClip_stream_typeは、Clip AV streamを示していなければならない。

SubPath_typeの８ビットのフィールドは、sub pathのタイプを示す。ここでは、図４１に示すように、'0x00'しか設定されておらず、他の値は、将来のために確保されている。

sync_PlayItem_idの８ビットのフィールドは、main pathの時間軸上でsub pathが再生開始する時刻が含まれるPlayItemのPlayItem_idを示す。所定のPlayItemに対応するPlayItem_idの値は、PlayList()において定義される（図２５参照）。

sync_start_PTS_of_PlayItemの３２ビットのフィールドは、main pathの時間軸上でsub pathが再生開始する時刻を示し、sync_PlayItem_idで参照されるPlayItem上のPTS (Presentaiotn Time Stamp) の上位３２ビットを示す。SubPath_IN_timeの３２ビットフィールドは、sub pathの再生開始時刻をストアする。SubPath_IN_timeは、sub pathの中で最初のプレゼンテーションユニットに対応する３３ビット長のPTSの上位３２ビットを示す。

SubPath_OUT_timeの３２ビットフィールドは、sub pathの再生終了時刻をストアする。SubPath_OUT_timeは、次式によって算出されるPresenation_end_TSの値の上位３２ビットを示す。

Presentation_end_TS = PTS_out + AU_duration

ここで、PTS_outは、SubPathの最後のプレゼンテーションユニットに対応する３３ビット長のPTSである。AU_durationは、SubPathの最後のプレゼンテーションユニットの９０kHz単位の表示期間である。

次に、図２３に示したxxxxx.rplsとyyyyy.vplsのシンタクス内のPlayListMark

()について説明する。PlayListについてのマーク情報は、このPlayListMarkにストアされる。図４２は、PlayListMarkのシンタクスを示す図である。図４２に示したPlayListMarkのシンタクスについて説明するに、version_numberは、このPlayListMark()のバージョンナンバを示す４個のキャラクタ文字である。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。

lengthは、このlengthフィールドの直後からPlayListMark()の最後までのPlayListMark()のバイト数を示す３２ビットの符号なし整数である。number_of_PlayList_marksは、PlayListMarkの中にストアされているマークの個数を示す１６ビットの符号なし整数である。number_of_PlayList_marksは、０であってもよい。mark_typeは、マークのタイプを示す８ビットのフィールドであり、図４３に示すテーブルに従って符号化される。

mark_time_stampの３２ビットフィールドは、マークが指定されたポイントを示すタイムスタンプをストアする。mark_time_stampのセマンティクスは、図４４に示すように、PlayList()において定義されるCPI_typeによって異なる。PlayItem_idは、マークが置かれているところのPlayItemを指定する８ビットのフィールドである。所定のPlayItemに対応するPlayItem_idの値は、PlayList()において定義される（図２５参照）。

character_setの８ビットのフィールドは、mark_nameフィールドに符号化されているキャラクタ文字の符号化方法を示す。その符号化方法は、図１９に示した値に対応する。name_lengthの８ビットフィールドは、mark_nameフィールドの中に示されるマーク名のバイト長を示す。mark_nameのフィールドは、マークの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクタ文字であり、それはマークの名称を示す。mark_nameフィールドの中で、それら有効なキャラクタ文字の後の値は、どのような値が設定されてもよい。

ref_thumbnail_indexのフィールドは、マークに付加されるサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのマークにはサムネイル画像が付加されており、そのサムネイル画像は、mark.thmbファイルの中にストアされている。その画像は、mark.thmbファイルの中でref_thumbnail_indexの値を用いて参照される（後述）。ref_thumbnail_indexフィールドが、

0xFFFFである場合、そのマークにはサムネイル画像が付加されていないことを示す。

次に、Clip Information fileについて説明する。zzzzz.clpi (Clip Information fileファイル) は、図４５に示すように６個のオブジェクトから構成される。それらは、ClipInfo()、STC_Info()、ProgramInfo()、CPI()、ClipMark()及びMakersPrivateData()である。AVストリーム (Clip AVストリーム又はBridge-Clip AV stream) とそれに対応するClip Informationファイルは、同じ数字列の"zzzzz"が使用される。

図４５に示したzzzzz.clpi (Clip Information fileファイル) のシンタクスについて説明するに、ClipInfo_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、ClipInfo()の先頭アドレスを示す。相対バイト数は０からカウントされる。

STC_Info_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、STC_Info()の先頭アドレスを示す。相対バイト数は０からカウントされる。ProgramInfo_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、ProgramInfo()の先頭アドレスを示す。相対バイト数は０からカウントされる。CPI_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、CPI()の先頭アドレスを示す。相対バイト数は０からカウントされる。

ClipMark_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、ClipMark()の先頭アドレスを示す。相対バイト数は０からカウントされる。MakersPrivateData_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、MakersPrivateData()の先頭アドレスを示す。相対バイト数は０からカウントされる。padding_word (パディングワード) は、zzzzz.clpiファイルのシンタクスに従って挿入される。Ｎ１、Ｎ２、Ｎ３、Ｎ４及びＮ５は、０又は任意の正の整数でなければならない。それぞれのパディングワードは、任意の値がとられるようにしてもよい。

次に、ClipInfoについて説明する。図４６は、ClipInfoのシンタクスを示す図である。ClipInfo()は、それに対応するAVストリームファイル (Clip AVストリー

ム又はBridge-Clip AVストリームファイル）の属性情報をストアする。

図４６に示したClipInfoのシンタクスについて説明するに、version_numberは、このClipInfo()のバージョンナンバを示す４個のキャラクタ文字である。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。lengthは、このlengthフィールドの直後からClipInfo()の最後までのClipInfo()のバイト数を示す３２ビットの符号なし整数である。Clip_stream_typeの８ビットのフィールドは、図４７に示すように、Clip Informationファイルに対応するAVストリームのタイプを示す。それぞれのタイプのAVストリームのストリームタイプについては後述する。

offset_SPNの３２ビットのフィールドは、AVストリーム（Clip AVストリーム又はBridge-Clip AVストリーム）ファイルの最初のソースパケットについてのソースパケット番号のオフセット値を与える。AVストリームファイルが最初にディスクに記録されるとき、このoffset_SPNは０でなければならない。

図４８に示すように、AVストリームファイルのはじめの部分が編集によって消去されたとき、offset_SPNは、０以外の値をとってもよい。本実施の形態では、offset_SPNを参照する相対ソースパケット番号（相対アドレス）が、しばしば、RSPN_xxx（xxxは変形する。例えば、RSPN_EP_start）の形式でシンタクスの中に記述されている。相対ソースパケット番号は、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからoffset_SPNの値を初期値としてカウントされる。

AVストリームファイルの最初のソースパケットから相対ソースパケット番号で参照されるソースパケットまでのソースパケットの数（SPN_xxx）は、次式で算出される。

SPN_xxx = RSPN_xxx - offset_SPN

図４８に、offset_SPNが、４である場合の例を示す。

TS_recording_rateは、２４ビットの符号なし整数であり、この値は、DVRドライブ（書込部２２）へ又はDVRドライブ（読出部２８）からのAVストリームの必要な入出力のビットレートを与える。record_time_and_dateは、Clipに対応するAVストリームが記録されたときの日時をストアする５６ビットのフィールドであり、

年／月／日／時／分／秒について、１４個の数字を４ビットのBinary Coded Decimal（BCD）で符号化したものである。例えば、2001/12/23:01:02:03は、"0x20011223010203"と符号化される。

durationは、Clipの総再生時間をアライバルタイムクロックに基づいた時間／分／秒の単位で示した２４ビットのフィールドである。このフィールドは、６個の数字を４ビットのBinary Coded Decimal（BCD）で符号化したものである。例えば、01:45:30は、"0x014530"と符号化される。

time_controlled_flagのフラグは、AVストリームファイルの記録モードを示す。このtime_controlled_flagが１である場合、記録モードは、記録してからの時間経過に対してファイルサイズが比例するようにして記録されるモードであることを示し、次式に示す条件を満たさなければならない。

$$TS\_average\_rate*192/188*(t - start\_time) - \alpha <= size\_clip(t) <= TS\_average\_rate*192/188*(t - start\_time) + \alpha$$

ここで、TS_average_rateは、AVストリームファイルのトランスポートストリームの平均ビットレートをbytes/secondの単位で表したものである。

また、上式において、ｔは、秒単位で表される時間を示し、start_timeは、AVストリームファイルの最初のソースパケットが記録されたときの時刻であり、秒単位で表される。size_clip(t)は、時刻ｔにおけるAVストリームファイルのサイズをバイト単位で表したものであり、例えば、start_timeから時刻tまでに１０個のソースパケットが記録された場合、size_clip(t)は１０＊１９２バイトである。αは、TS_average_rateに依存する定数である。

time_controlled_flagが０にセットされている場合、記録モードは、記録の時間経過とAVストリームのファイルサイズが比例するように制御していないことを示す。例えば、これは入力トランスポートストリームをトランスペアレント記録する場合である。

TS_average_rateは、time_controlled_flagが１にセットされている場合、この２４ビットのフィールドは、上式で用いているTS_average_rateの値を示す。time_controlled_flagが０にセットされている場合、このフィールドは、何も意味を持たず、０にセットされなければならない。例えば、可変ビットレートのトラン

スポートストリームは、次に示す手順により符号化される。まずトランスポートレートをTS_recording_rateの値にセットする。次に、ビデオストリームを可変ビットレートで符号化する。そして、ヌルパケットを使用しないことによって、間欠的にトランスポートパケットを符号化する。

RSPN_arrival_time_discontinuityの３２ビットフィールドは、Bridge-Clip AV streamファイル上でアライバルタイムベースの不連続が発生する場所の相対アドレスである。RSPN_arrival_time_discontinuityは、ソースパケット番号を単位とする大きさであり、Bridge-Clip AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのBridge-Clip AV streamファイルの中での絶対アドレスは、上述した

SPN_xxx = RSPN_xxx - offset_SPN

に基づいて算出される。

reserved_for_system_useの１４４ビットのフィールドは、システム用にリザーブされている。is_format_identifier_validのフラグが１であるとき、format_identifierのフィールドが有効であることを示す。is_original_network_ID_validのフラグが１である場合、original_network_IDのフィールドが有効であることを示す。is_transport_stream_ID_validのフラグが１である場合、transport_stream_IDのフィールドが有効であることを示す。is_servece_ID_validのフラグが１である場合、servece_IDのフィールドが有効であることを示す。

is_country_code_validのフラグが１であるとき、country_codeのフィールドが有効であることを示す。format_identifierの３２ビットフィールドは、トランスポートストリームの中でregistration deascriotor（ISO/IEC13818-1で定義されている）が持つformat_identifierの値を示す。original_network_IDの１６ビットフィールドは、トランスポートストリームの中で定義されているoriginal_network_IDの値を示す。transport_stream_IDの１６ビットフィールドは、トランスポートストリームの中で定義されているtransport_stream_IDの値を示す。

servece_IDの１６ビットフィールドは、トランスポートストリームの中で定義されているservece_IDの値を示す。country_codeの２４ビットのフィールドは、ISO3166によって定義されるカントリーコードを示す。それぞれのキャラクタ文字

は、ISO8859-1で符号化される。例えば、日本は"JPN"と表され、"0x4A 0x50 0x4E"と符号化される。stream_format_nameは、トランスポートストリームのストリーム定義をしているフォーマット機関の名称を示すISO-646の１６個のキャラクタコードである。このフィールドの中の無効なバイトは、値'0xFF'がセットされる。

format_identifier、original_network_ID、transport_stream_ID、servece_ID、country_code及びstream_format_nameは、トランスポートストリームのサービスプロバイダを示すものであり、これにより、オーディオやビデオストリームの符号化制限、SI（サービスインフォメーション）の規格やオーディオビデオストリーム以外のプライベートデータストリームのストリーム定義を認識することができる。これらの情報は、デコーダが、そのストリームをデコードできるか否か、そしてデコードできる場合にデコード開始前にデコーダシステムの初期設定を行うために用いることが可能である。

次に、STC_Infoについて説明する。ここでは、MPEG-2トランスポートストリームの中でSTCの不連続点（システムタイムベースの不連続点）を含まない時間区間をSTC_sequenceと称し、Clipの中で、STC_sequenceは、STC_sequence_idの値によって特定される。図５０Ａ，図５０Ｂは、連続なSTC区間について説明する図である。同じSTC_sequenceの中で同じSTCの値は、決して現れない（ただし、後述するように、Clipの最大時間長は制限されている）。したがって、同じSTC_sequenceの中で同じPTSの値もまた、決して現れない。AVストリームが、N（N>0）個のSTC不連続点を含む場合、Clipのシステムタイムベースは、(N+1)個のSTC_sequenceに分割される。

STC_Infoは、STCの不連続（システムタイムベースの不連続）が発生する場所のアドレスをストアする。図５１を参照して説明するように、RSPN_STC_startが、そのアドレスを示し、最後のSTC_sequenceを除くk番目（k>=0）のSTC_sequenceは、k番目のRSPN_STC_startで参照されるソースパケットが到着した時刻から始まり、(k+1)番目のRSPN_STC_startで参照されるソースパケットが到着した時刻で終わる。最後のSTC_sequenceは、最後のRSPN_STC_startで参照されるソースパケットが到着した時刻から始まり、最後のソースパケットが到着した時刻で終了する。

図５２は、STC_Infoのシンタクスを示す図である。図５２に示したSTC_Infoの

シンタクスについて説明するに、version_numberは、このSTC_Info()のバージョンナンバを示す４個のキャラクタ文字である。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。

lengthは、このlengthフィールドの直後からSTC_Info()の最後までのSTC_Info()のバイト数を示す３２ビットの符号なし整数である。CPI()のCPI_typeがTU_map typeを示す場合、このlengthフィールドは０をセットしてもよい。CPI()のCPI_typeがEP_map typeを示す場合、num_of_STC_sequencesは１以上の値でなければならない。

num_of_STC_sequencesの８ビットの符号なし整数は、Clipの中でのSTC_sequenceの数を示す。この値は、このフィールドに続くfor-loopのループ回数を示す。所定のSTC_sequenceに対応するSTC_sequence_idは、RSPN_STC_startを含むfor-loopの中で、そのSTC_sequenceに対応するRSPN_STC_startの現れる順番により定義されるものである。STC_sequence_idは、０から開始される。

RSPN_STC_startの３２ビットフィールドは、AVストリームファイル上でSTC_sequenceが開始するアドレスを示す。RSPN_STC_startは、AVストリームファイルの中でシステムタイムベースの不連続点が発生するアドレスを示す。RSPN_STC_startは、AVストリームの中で新しいシステムタイムベースの最初のPCRを持つソースパケットの相対アドレスとしてもよい。RSPN_STC_startは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAV streamファイルの中での絶対アドレスは、既に上述した

SPN_xxx = RSPN_xxx - offset_SPN

により算出される。

次に、図４５に示したzzzzz.clipのシンタクス内のProgramInfoについて説明する。図５３を参照しながら説明するに、ここでは、Clipの中で次の特徴をもつ時間区間をprogram_sequenceと呼ぶ。まず、PCR_PIDの値が変わらない。次に、ビデオエレメンタリストリームの数が変化しない。また、それぞれのビデオストリームについてのPIDの値とそのVideoCodingInfoによって定義される符号化情報が変化しない。さらに、オーディオエレメンタリストリームの数が変化しない。また、

それぞれのオーディオストリームについてのPIDの値とそのAudioCodingInfoによって定義される符号化情報が変化しない。

program_sequenceは、同一の時刻において、ただ１つのシステムタイムベースを持つ。program_sequenceは、同一の時刻において、ただ１つのPMTを持つ。ProgramInfo()は、program_sequenceが開始する場所のアドレスをストアする。RSPN_program_sequence_startが、そのアドレスを示す。

図５４は、ProgramInfoのシンタクスを示す図である。図５４に示したProgramInfoのシンタクスを説明するに、version_numberは、このProgramInfo()のバージョンナンバを示す４個のキャラクタ文字である。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。

lengthは、このlengthフィールドの直後からProgramInfo()の最後までのProgramInfo()のバイト数を示す３２ビットの符号なし整数である。CPI()のCPI_typeがTU_map typeを示す場合、このlengthフィールドは０にセットされてもよい。CPI()のCPI_typeがEP_map typeを示す場合、number_of_programsは１以上の値でなければならない。

number_of_program_sequencesの８ビットの符号なし整数は、Clipの中でのprogram_sequenceの数を示す。この値は、このフィールドに続くfor-loopのループ回数を示す。Clipの中でprogram_sequenceが変化しない場合、number_of_program_sequencesは１をセットされなければならない。RSPN_program_sequence_startの３２ビットフィールドは、AVストリームファイル上でプログラムシーケンスが開始する場所の相対アドレスである。

RSPN_program_sequence_startは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAVストリームファイルの中での絶対アドレスは、

SPN_xxx = RSPN_xxx - offset_SPN

により算出される。シンタクスのfor-loopの中でRSPN_program_sequence_start値は、昇順に現れなければならない。

PCR_PIDの１６ビットフィールドは、そのprogram_sequenceに有効なPCRフィー

ルドを含むトランスポートパケットのPIDを示す。number_of_videosの８ビットフィールドは、video_stream_PIDとVideoCodingInfo()を含むfor-loopのループ回数を示す。number_of_audiosの８ビットフィールドは、audio_stream_PIDとAudioCodingInfo()を含むfor-loopのループ回数を示す。video_stream_PIDの１６ビットフィールドは、そのprogram_sequenceに有効なビデオストリームを含むトランスポートパケットのPIDを示す。このフィールドに続くVideoCodingInfo()は、そのvideo_stream_PIDで参照されるビデオストリームの内容を説明しなければならない。

audio_stream_PIDの１６ビットフィールドは、そのprogram_sequenceに有効なオーディオストリームを含むトランスポートパケットのPIDを示す。このフィールドに続くAudioCodingInfo()は、そのaudio_stream_PIDで参照されるビデオストリームの内容を説明しなければならない。

なお、シンタクスのfor-loopの中でvideo_stream_PIDの値の現れる順番は、そのprogram_sequenceに有効なPMTの中でビデオストリームのPIDが符号化されている順番に等しくなければならない。また、シンタクスのfor-loopの中でaudio_stream_PIDの値の現れる順番は、そのprogram_sequenceに有効なPMTの中でオーディオストリームのPIDが符号化されている順番に等しくなければならない。

図５５は、図５４に示したProgramInfoのシンタクス内のVideoCodingInfoのシンタクスを示す図である。図５５に示したVideoCodingInfoのシンタクスを説明するに、video_formatの８ビットフィールドは、図５６に示すように、ProgramInfo()の中のvideo_stream_PIDに対応するビデオフォーマットを示す。

frame_rateの８ビットフィールドは、図５７に示すように、ProgramInfo()の中のvideo_stream_PIDに対応するビデオのフレームレートを示す。display_aspect_ratioの８ビットフィールドは、図５８に示すように、ProgramInfo()の中のvideo_stream_PIDに対応するビデオの表示アスペクト比を示す。

図５９は、図５４に示したProgramInfoのシンタクス内のAudioCodingInfoのシンタクスを示す図である。図５９に示したAudioCodingInfoのシンタクスを説明するに、audio_codingの８ビットフィールドは、図６０に示すように、ProgramInfo()の中のaudio_stream_PIDに対応するオーディオの符号化方法を示す。

audio_component_typeの８ビットフィールドは、図６１に示すように、ProgramInfo()の中のaudio_stream_PIDに対応するオーディオのコンポーネントタイプを示す。sampling_frequencyの８ビットフィールドは、図６２に示すように、ProgramInfo()の中のaudio_stream_PIDに対応するオーディオのサンプリング周波数を示す。

次に、図４５に示したzzzzz.clipのシンタクス内のCPI (Characteristic Point Information) について説明する。CPIは、AVストリームの中の時間情報とそのファイルの中のアドレスとを関連付けるためにある。CPIには２つのタイプがあり、それらはEP_mapとTU_mapである。図６３に示すように、CPI()の中のCPI_typeがEP_map typeの場合、そのCPI()はEP_mapを含む。図６４に示すように、CPI()の中のCPI_typeがTU_map typeの場合、そのCPI()はTU_mapを含む。１つのAVストリームは、１つのEP_map又は１つのTU_mapを持つ。AVストリームがSESFトランスポートストリームの場合、それに対応するClipはEP_mapを持たなければならない。

図６５は、CPIのシンタクスを示す図である。図６５に示したCPIのシンタクスを説明するに、version_numberは、このCPI()のバージョンナンバを示す４個のキャラクタ文字である。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。lengthは、このlengthフィールドの直後からCPI()の最後までのCPI()のバイト数を示す３２ビットの符号なし整数である。CPI_typeは、図６６に示すように、１ビットのフラグであり、ClipのCPIのタイプを表す。

次に、図６５に示したCPIのシンタクス内のEP_mapについて説明する。EP_mapには、２つのタイプがあり、それはビデオストリーム用のEP_mapとオーディオストリーム用のEP_mapである。EP_mapの中のEP_map_typeが、EP_mapのタイプを区別する。Clipが１つ以上のビデオストリームを含む場合、ビデオストリーム用のEP_mapが使用されなければならない。Clipがビデオストリームを含まず、１つ以上のオーディオストリームを含む場合、オーディオストリーム用のEP_mapが使用されなければならない。

ビデオストリーム用のEP_mapについて図６７を参照して説明する。ビデオストリーム用のEP_mapは、stream_PID、PTS_EP_start及びRSPN_EP_startというデータを持つ。stream_PIDは、ビデオストリームを伝送するトランスポートパケットの

PIDを示す。PTS_EP_startは、ビデオストリームのシーケンスヘッダから始まるアクセスユニットのPTSを示す。RSPN_EP_startは、AVストリームの中でPTS_EP_startにより参照されるアクセスユニットの第１バイト目を含むソースパケットのアドレスを示す。

EP_map_for_one_stream_PID()と呼ばれるサブテーブルは、同じPIDを持つトランスポートパケットによって伝送されるビデオストリーム毎に作られる。Clipの中に複数のビデオストリームが存在する場合、EP_mapは複数のEP_map_for_one_stream_PID()を含んでもよい。

オーディオストリーム用のEP_mapは、stream_PID、PTS_EP_start及びRSPN_EP_startというデータを持つ。stream_PIDは、オーディオストリームを伝送するトランスポートパケットのPIDを示す。PTS_EP_startは、オーディオストリームのアクセスユニットのPTSを示す。RSPN_EP_startは、AVストリームの中でPTS_EP_startで参照されるアクセスユニットの第１バイト目を含むソースパケットのアドレスを示す。

EP_map_for_one_stream_PID()と呼ばれるサブテーブルは、同じPIDを持つトランスポートパケットによって伝送されるオーディオストリーム毎に作られる。Clipの中に複数のオーディオストリームが存在する場合、EP_mapは複数のEP_map_for_one_stream_PID()を含んでもよい。

EP_mapとSTC_Infoの関係を説明するに、１つのEP_map_for_one_stream_PID()は、STCの不連続点に関係なく１つのテーブルに作られる。RSPN_EP_startの値とSTC_Info()において定義されるRSPN_STC_startの値を比較することにより、それぞれのSTC_sequenceに属するEP_mapのデータの境界が分かる（図６８を参照）。EP_mapは、同じPIDで伝送される連続したストリームの範囲に対して、１つのEP_map_for_one_stream_PIDを持たねばならない。図６９に示したような場合、program#1とprogram#3は、同じビデオPIDを持つが、データ範囲が連続していないので、それぞれのプログラム毎にEP_map_for_one_stream_PIDを持たねばならない。

図７０は、EP_mapのシンタクスを示す図である。図７０に示したEP_mapのシンタクスを説明するに、EP_typeは、４ビットのフィールドであり、図７１に示すように、EP_mapのエントリポイントタイプを示す。EP_typeは、このフィールドに続

くデータフィールドのセマンティクスを示す。Clipが１つ以上のビデオストリームを含む場合、EP_typeは０（'video'）にセットされなければならない。あるいはClipがビデオストリームを含まず、１つ以上のオーディオストリームを含む場合、EP_typeは１（'audio'）にセットされなければならない。

number_of_stream_PIDsの１６ビットのフィールドは、EP_map()の中のnumber_of_stream_PIDsを変数にもつfor-loopのループ回数を示す。stream_PID(k)の１６ビットのフィールドは、EP_map_for_one_stream_PID(num_EP_entries(k))によって参照されるk番目のエレメンタリストリーム（ビデオ又はオーディオストリーム）を伝送するトランスポートパケットのPIDを示す。EP_typeが０（'video'）に等しい場合、そのエレメンタリストリームはビデオストリームでなければならない。また、EP_typeが１（'audio'）に等しい場合、そのエレメンタリストリームはオーディオストリームでなければならない。

num_EP_entries(k)の１６ビットのフィールドは、EP_map_for_one_stream_PID(num_EP_entries(k))によって参照されるnum_EP_entries(k)を示す。EP_map_for_one_stream_PID_Start_address(k)：この３２ビットのフィールドは、EP_map()の中でEP_map_for_one_stream_PID(num_EP_entries(k))が始まる相対バイト位置を示す。この値は、EP_map()の第１バイト目からの大きさで示される。

padding_wordは、EP_map()のシンタクスに従って挿入されなければならない。XとYは、０又は任意の正の整数でなければならない。それぞれのパディングワードは、任意の値を取ってもよい。

図７２は、EP_map_for_one_stream_PIDのシンタクスを示す図である。図７２に示したEP_map_for_one_stream_PIDのシンタクスを説明するに、PTS_EP_startの３２ビットのフィールドのセマンティクスは、EP_map()において定義されるEP_typeにより異なる。EP_typeが０（'video'）に等しい場合、このフィールドは、ビデオストリームのシーケンスヘッダで始まるアクセスユニットの３３ビット精度のPTSの上位３２ビットを持つ。EP_typeが１（'audio'）に等しい場合、このフィールドは、オーディオストリームのアクセスユニットの３３ビット精度のPTSの上位３２ビットを持つ。

RSPN_EP_startの３２ビットのフィールドのセマンティクスは、EP_map()におい

て定義されるEP_typeにより異なる。EP_typeが０（'video'）に等しい場合、このフィールドは、AVストリームの中でPTS_EP_startにより参照されるアクセスユニットのシーケンスヘッダの第１バイト目を含むソースパケットの相対アドレスを示す。あるいはEP_typeが１（'audio'）に等しい場合、このフィールドは、AVストリームの中でPTS_EP_startにより参照されるアクセスユニットのオーディオフレームの第１バイト目を含むソースパケットの相対アドレスを示す。

RSPN_EP_startは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAVストリームファイルの中での絶対アドレスは、

SPN_xxx = RSPN_xxx - offset_SPN

により算出される。シンタクスのfor-loopの中でRSPN_EP_startの値は、昇順に現れなければならない。

次に、TU_mapについて、図７３を参照して説明する。TU_mapは、ソースパケットのアライバルタイムクロック（到着時刻ベースの時計）に基づいて、１つの時間軸を作る。その時間軸は、TU_map_time_axisと呼ばれる。TU_map_time_axisの原点は、TU_map()の中のoffset_timeによって示される。TU_map_time_axisは、offset_timeから一定の単位に分割される。その単位を、time_unitと称する。

AVストリームの中の各々のtime_unitの中で、最初の完全な形のソースパケットのAVストリームファイル上のアドレスが、TU_mapにストアされる。これらのアドレスを、RSPN_time_unit_startと称する。TU_map_time_axis上において、k（k>=0）番目のtime_unitが始まる時刻は、TU_start_time(k)と呼ばれる。この値は次式に基づいて算出される。

TU_start_time(k) = offset_time + k*time_unit_size

TU_start_time(k)は、４５kHzの精度を持つ。

図７４は、TU_mapのシンタクスを示す図である。図７４に示したTU_mapのシンタクスを説明するに、offset_timeの３２ビット長のフィールドは、TU_map_time_axisに対するオフセットタイムを与える。この値は、Clipの中の最初のtime_unitに対するオフセット時刻を示す。offset_timeは、２７MHz精度のアライバルタ

イムクロックから導き出される４５kHzクロックを単位とする大きさである。AVストリームが新しいClipとして記録される場合、offset_timeは０にセットされなければならない。

time_unit_sizeの３２ビットフィールドは、time_unitの大きさを与えるものであり、それは２７MHz精度のアライバルタイムクロックから導き出される４５kHzクロックを単位とする大きさである。time_unit_sizeは、１秒以下（time_unit_size<=４５０００）にすることがよい。number_of_time_unit_entriesの３２ビットフィールドは、TU_map()の中にストアされているtime_unitのエントリ数を示す。

RSPN_time_unit_startの３２ビットフィールドは、AVストリームの中でそれぞれのtime_unitが開始する場所の相対アドレスを示す。RSPN_time_unit_startは、ソースパケット番号を単位とする大きさであり、AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAV streamファイルの中での絶対アドレスは、

SPN_xxx = RSPN_xxx - offset_SPN

により算出される。シンタクスのfor-loopの中でRSPN_time_unit_startの値は、昇順に現れなければならない。(k+1)番目のtime_unitの中にソースパケットが何もない場合、(k+1)番目のRSPN_time_unit_startは、k番目のRSPN_time_unit_startと等しくなければならない。

図４５に示したzzzzz.clipのシンタクス内のClipMarkについて説明する。ClipMarkは、クリップについてのマーク情報であり、ClipMarkの中にストアされる。このマークは、記録器（記録再生装置１）によってセットされるものであり、ユーザによってセットされるものではない。

図７５は、ClipMarkのシンタクスを示す図である。図７５に示したClipMarkのシンタクスを説明するに、version_numberは、このClipMark()のバージョンナンバを示す４個のキャラクタ文字である。version_numberは、ISO 646に従って、”0045”と符号化されなければならない。

lengthは、このlengthフィールドの直後からClipMark()の最後までのClipMark()のバイト数を示す３２ビットの符号なし整数である。number_of_Clip_marksは、ClipMarkの中にストアされているマークの個数を示す１６ビットの符号なし整数。

number_of_Clip_marksは、０であってもよい。mark_typeは、マークのタイプを示す８ビットのフィールドであり、図７６に示すテーブルに従って符号化される。

mark_time_stampは、３２ビットフィールドであり、マークが指定されたポイントを示すタイムスタンプをストアする。mark_time_stampのセマンティクスは、図７７に示すように、PlayList()の中のCPI_typeにより異なる。

STC_sequence_idは、CPI()の中のCPI_typeがEP_map typeを示す場合、この８ビットのフィールドは、mark_time_stampが置かれているところのSTC連続区間のSTC_sequence_idを示す。CPI()の中のCPI_typeがTU_map typeを示す場合、この８ビットのフィールドは何も意味を持たず、０にセットされる。character_setの８ビットのフィールドは、mark_nameフィールドに符号化されているキャラクタ文字の符号化方法を示す。その符号化方法は、図１９に示される値に対応する。

name_lengthの８ビットフィールドは、mark_nameフィールドの中に示されるマーク名のバイト長を示す。mark_nameのフィールドは、マークの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクタ文字であり、それはマークの名称を示す。mark_nameフィールドの中で、それら有効なキャラクタ文字の後の値は、どんな値が入っていてもよい。

ref_thumbnail_indexのフィールドは、マークに付加されるサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのマークにはサムネイル画像が付加されており、そのサムネイル画像は、mark.thmbファイルの中にストアされている。その画像は、mark.thmbファイルの中でref_thumbnail_indexの値を用いて参照される。ref_thumbnail_indexフィールドが、0xFFFFである場合、そのマークにはサムネイル画像が付加されていない。

図７８は、図７５に代わるClipMarkの他のシンタクスを示す図であり、図７９は、その場合における、図７６に代わるmark_typeのテーブルの例を示す。reserved_for_maker_IDは、mark_typeが、0xC0から0xFFの値を示すときに、そのmark_typeを定義しているメーカのメーカＩＤを示す１６ビットのフィールドである。メーカＩＤは、DVRフォーマットライセンサが指定する。mark_entry()は、マーク点に指定されたポイントを示す情報であり、そのシンタクスの詳細は後述する。representative_picture_entry()は、mark_entry()によって示されるマークを代

表する画像のポイントを示す情報であり、そのシンタクスの詳細は後述する。

ClipMarkは、ユーザがAVストリームを再生するときに、その内容を視覚的に検索できるようにするために用いられる。DVRプレーヤは、GUI（グラフィカルユーザインタフェース）を使用して、ClipMarkの情報をユーザに提示する。ClipMarkの情報を視覚的に表示するためには、mark_entry()が示すピクチャよりもむしろrepresentative_picture_entry()が示すピクチャを示した方がよい。

図８０に、mark_entry()とrepresentative_picture_entry()の例を示す。例えば、あるプログラムが開始してから、しばらくした後（数秒後）、そのプログラムの番組名（タイトル）が表示されるとする。ClipMarkを作るときは、mark_entry()は、そのプログラムの開始ポイントに置き、representative_picture_entry()は、そのプログラムの番組名（タイトル）が表示されるポイントに置くようにしてもよい。

DVRプレーヤは、representative_picture_entryの画像をGUIに表示し、ユーザがその画像を指定すると、DVRプレーヤは、mark_entryの置かれたポイントから再生を開始する。

mark_entry()及びrepresentative_picture_entry()のシンタクスを、図８１に示す。

mark_time_stampは、３２ビットフィールドであり、mark_entry()の場合はマークが指定されたポイントを示すタイムスタンプをストアし、またrepresentative_picture_entry()の場合、mark_entry()によって示されるマークを代表する画像のポイントを示すタイムスタンプをストアする。

次に、ClipMarkを指定するために、PTSによるタイムスタンプベースの情報を使用するのではなく、アドレスベースの情報を使用する場合のmark_entry()とrepresentative_picture_entry()のシンタクスの例を図８２に示す。

RSPN_ref_EP_startは、mark_entry()の場合、AVストリームの中でマーク点のピクチャをデコードするためのストリームのエントリポイントを示すソースパケットの相対アドレスを示す。また、representative_picture_entry()の場合、mark_entry()によって示されるマークを代表するピクチャをデコードするためのストリームのエントリポイントを示すソースパケットの相対アドレスを示す。RSPN_r

ef_EP_startの値は、EP_mapの中にRSPN_EP_startとしてストアされていなければならず、かつ、そのRSPN_EP_startに対応するPTS_EP_startの値は、EP_mapの中で、マーク点のピクチャのPTSより過去で最も近い値でなければならない。

offset_num_picturesは、３２ビットのフィールドであり、RSPN_ref_EP_startにより参照されるピクチャから表示順序でマーク点で示されるピクチャまでのオフセットのピクチャ数を示す。この数は、０からカウントされる。図８３の例の場合、offset_num_picturesは６となる。

次に、ClipMarkを指定するために、アドレスベースの情報を使用する場合のmark_entry()とrepresentative_picture_entry()のシンタクスの別の例を図８４に示す。

RSPN_mark_pointは、mark_entry()の場合、AVストリームの中で、そのマークが参照するアクセスユニットの第１バイト目を含むソースパケットの相対アドレスを示す。また、representative_picture_entry()の場合、mark_entry()によって示されるマークを代表する符号化ピクチャの第１バイト目を含むソースパケットの相対アドレスを示す。

RSPN_mark_pointは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClip Information fileにおいて定義されるoffset_SPNの値を初期値としてカウントされる。

図８５を用いて、ClipMarkとEP_mapの関係を説明する。この例の場合、EP_mapが、エントリポイントのアドレスとしてI0、I1、Inを指定しており、これらのアドレスからシーケンスヘッダに続くIピクチャが開始しているとする。ClipMarkが、あるマークのアドレスとして、M1を指定しているとき、そのソースパケットから開始しているピクチャをデコードできるためには、M1のアドレスより前で最も近いエントリポイントであるI1からデータを読み出し開始すればよい。

MakersPrivateDataについては、図２２を参照して既に説明したので、その説明は省略する。

次に、サムネイルインフォメーション（Thumbnail Information）について説明する。サムネイル画像は、menu.thmbファイル又はmark.thmbファイルにストアされる。これらのファイルは同じシンタクス構造であり、ただ１つのThumbnail()を

持つ。menu.thmbファイルは、メニューサムネイル画像、すなわちVolumeを代表する画像、及びそれぞれのPlayListを代表する画像をストアする。全てのメニューサムネイルは、ただ１つのmenu.thmbファイルにストアされる。

mark.thmbファイルは、マークサムネイル画像、すなわちマーク点を表すピクチャをストアする。全てのPlayList及びClipに対する全てのマークサムネイルは、ただ１つのmark.thmbファイルにストアされる。サムネイルは頻繁に追加、削除されるので、追加操作と部分削除の操作は容易に高速に実行できなければならない。この理由のため、Thumbnail()はブロック構造を有する。画像のデータは幾つかの部分に分割され、各部分は１つのtn_blockに格納される。１つの画像データは連続したtn_blockに格納される。tn_blockの列には、使用されていないtn_blockが存在してもよい。１つのサムネイル画像のバイト長は可変である。

図８６は、menu.thmbとmark.thmbのシンタクスを示す図であり、図８７は、図８６に示したmenu.thmbとmark.thmbのシンタクス内のThumbnailのシンタクスを示す図である。図８７に示したThumbnailのシンタクスについて説明するに、version_numberは、このThumbnail()のバージョンナンバを示す４個のキャラクタ文字である。version_numberは、ISO 646に従って、”0045”と符号化されなければならない。

lengthは、このlengthフィールドの直後からThumbnail()の最後までのMakersPrivateData()のバイト数を示す３２ビットの符号なし整数である。tn_blocks_start_addressは、Thumbnail()の先頭のバイトからの相対バイト数を単位として、最初のtn_blockの先頭バイトアドレスを示す３２ビットの符号なし整数である。相対バイト数は０からカウントされる。number_of_thumbnailsは、Thumbnail()の中に含まれているサムネイル画像のエントリ数を与える１６ビットの符号なし整数である。

tn_block_sizeは、１０２４バイトを単位として、１つのtn_blockの大きさを与える１６ビットの符号なし整数である。例えば、tn_block_size＝１ならば、それは１つのtn_blockの大きさが１０２４バイトであることを示す。number_of_tn_blocksは、このThumbnail()中のtn_blockのエントリ数を表す１６ビットの符号なし整数である。thumbnail_indexは、このthumbnail_indexフィールドから始まる

forループ１回分のサムネイル情報で表されるサムネイル画像のインデックス番号を表す１６ビットの符号なし整数である。thumbnail_indexとして、0xFFFFという値を使用してはならない。thumbnail_indexはUIAppInfoVolume()、UIAppInfoPlayList()、PlayListMark()及びClipMark()の中のref_thumbnail_indexによって参照される。

thumbnail_picture_formatは、サムネイル画像のピクチャフォーマットを表す８ビットの符号なし整数で、図８８に示すような値をとる。表中のDCFとPNGは"menu.thmb"内でのみ許される。マークサムネイルは、値"0x00"（MPEG-2 Video I-picture）をとらなければならない。

picture_data_sizeは、サムネイル画像のバイト長をバイト単位で示す３２ビットの符号なし整数である。start_tn_block_numberは、サムネイル画像のデータが始まるtn_blockのtn_block番号を表す１６ビットの符号なし整数である。サムネイル画像データの先頭は、tb_blockの先頭と一致していなければならない。tn_block番号は、０から始まり、tn_blockのfor-loop中の変数kの値に関係する。

x_picture_lengthは、サムネイル画像のフレーム画枠の水平方向のピクセル数を表す１６ビットの符号なし整数である。y_picture_lengthは、サムネイル画像のフレーム画枠の垂直方向のピクセル数を表す１６ビットの符号なし整数である。tn_blockは、サムネイル画像がストアされる領域である。Thumbnail()の中の全てのtn_blockは、同じサイズ（固定長）であり、その大きさはtn_block_sizeによって定義される。

図８９Ａ，図８９Ｂは、サムネイル画像データがどのようにtn_blockに格納されるかを模式的に表した図である。図８９Ａ，図８９Ｂのように、各サムネイル画像データはtn_blockの先頭から始まり、1tn_blockを超える大きさの場合は、連続する次のtn_blockを使用してストアされる。このようにすることにより、可変長であるピクチャデータが、固定長のデータとして管理することが可能となり、削除といった編集に対して簡便な処理により対応することができるようになる。

次に、AVストリームファイルについて説明する。AVストリームファイルは、"M2TS"ディレクトリ（図１４）にストアされる。AVストリームファイルには、２つのタイプがあり、それらは、Clip AVストリームとBridge-Clip AVストリームファ

イルである。両方のAVストリーム共に、これ以降で定義されるDVR MPEG-2トランスポートストリームファイルの構造でなければならない。

まず、DVR MPEG-2トランスポートストリームについて説明する。DVR MPEG-2トランスポートストリームの構造は、図９０に示すようになっている。AVストリームファイルは、DVR MPEG-2トランスポートストリームの構造を持つ。DVR MPEG-2トランスポートストリームは、整数個のAligned unitから構成される。Aligned unitの大きさは、６１４４バイト（２０４８＊３バイト）である。Aligned unitは、ソースパケットの第１バイト目から始まる。ソースパケットは、１９２バイト長である。１つのソースパケットは、TP_extra_headerとトランスポートパケットから成る。TP_extra_headerは、４バイト長であり、またトランスポートパケットは、１８８バイト長である。

１つのAligned unitは、３２個のソースパケットから成る。DVR MPEG-2トランスポートストリームの中の最後のAligned unitも、また３２個のソースパケットから成る。よって、DVR MPEG-2トランスポートストリームは、Aligned unitの境界で終端する。ディスクに記録される入力トランスポートストリームのトランスポートパケットの数が３２の倍数でないとき、ヌルパケット（PID=0x1FFFのトランスポートパケット）を持ったソースパケットを最後のAligned unitに使用しなければならない。ファイルシステムは、DVR MPEG-2トランスポートストリームに余分な情報を付加してはならない。

図９１に、DVR MPEG-2トランスポートストリームのレコーダモデルを示す。図９１に示したレコーダは、レコーディングプロセスを規定するための概念上のモデルである。DVR MPEG-2トランスポートストリームは、このモデルに従う。

MPEG-2トランスポートストリームの入力タイミングについて説明する。入力MPEG-2トランスポートストリームは、フルトランスポートストリーム又はパーシャルトランスポートストリームである。入力されるMPEG-2トランスポートストリームは、ISO/IEC13818-1又はISO/IEC13818-9に従っていなければならない。MPEG-2トランスポートストリームのi番目のバイトは、T-STD（ISO/IEC 13818-1で規定されるTransport stream system target decoder）５１とソースパケッタイザ（sourse packetizer）５４へ、時刻t(i)に同時に入力される。Rpkは、トランスポー

トパケットの入力レートの瞬時的な最大値である。

２７MHz PLL５２は、２７MHzクロックの周波数を発生する。２７MHzクロックの周波数は、MPEG-2トランスポートストリームのPCR (Program Clock Reference) の値にロックされる。アライバルタイムクロックカウンタ (arrival time clock counter) ５３は、２７MHzの周波数のパルスをカウントするバイナリカウンタである。arrival_time_clock(i)は、時刻t(i)におけるarrival time clock counter５３のカウント値である。

source packetizer５４は、全てのトランスポートパケットにTP_extra_headerを付加し、ソースパケットを作る。arrival_time_stampは、トランスポートパケットの第１バイト目がT-STD５１とソースパケッタイザ５４の両方へ到着する時刻を表す。arrival_time_stamp(k)は、次式で示されるようにarrival_time_clock(k)のサンプル値であり、ここで、kはトランスポートパケットの第１バイト目を示す。

arrival_time_stamp(k) = arrival_time_clock(k)% 230

２つの連続して入力されるトランスポートパケットの時間間隔が、２３０／２７０００００００秒（約４０秒）以上になる場合、その２つのトランスポートパケットのarrival_time_stampの差分は、２３０／２７０００００００秒になるようにセットされるべきである。レコーダは、そのようになる場合に備えてある。

スムージングバッファ (smoothing buffer) ５５は、入力トランスポートストリームのビットレートをスムージングする。スムージングバッファ５５は、オーバーフローしてはならない。Rmaxは、スムージングバッファ５５が空でないときのスムージングバッファ５５からのソースパケットの出力ビットレートである。スムージングバッファ５５が空であるとき、スムージングバッファ５５からの出力ビットレートは０である。

次に、DVR MPEG-2トランスポートストリームのレコーダモデルのパラメータについて説明する。Rmaxという値は、AVストリームファイルに対応するClipInfo()において定義されるTS_recording_rateによって与えられる。この値は、次式により算出される。

Rmax = TS_recording_rate * 192/188

TS_recording_rateの値は、bytes/secondを単位とする大きさである。

入力トランスポートストリームがSESFトランスポートストリームの場合、Rpkは、AVストリームファイルに対応するClipInfo()において定義されるTS_recording_rateに等しくなければならない。入力トランスポートストリームがSESFトランスポートストリームでない場合、この値はMPEG-2 transport streamのデスクリプタ、例えばmaximum_bitrate_descriptorやpartial_transport_stream_descriptorなどにおいて定義される値を参照してもよい。

入力トランスポートストリームがSESFトランスポートストリームの場合、スムージングバッファ５５の大きさ（smoothing buffer size）は０である。入力トランスポートストリームがSESFトランスポートストリームでない場合、スムージングバッファ５５の大きさはMPEG-2 transport streamのデスクリプタ、例えばsmoothing_buffer_descriptor、short_smoothing_buffer_descriptor、partial_transport_stream_descriptorなどにおいて定義される値を参照してもよい。

記録機（レコーダ）及び再生機（プレーヤ）は、十分なサイズのバッファを用意しなければならない。ディフォルトのバッファサイズは、１５３６バイトである。

次に、DVR MPEG-2トランスポートストリームのプレーヤモデルについて説明する。図９２は、DVR MPEG-2トランスポートストリームのプレーヤモデルを示す図である。これは、再生プロセスを規定するための概念上のモデルである。DVR MPEG-2トランスポートストリームは、このモデルに従う。

27MHz X-tal（クリスタル発振器）６１は、２７MHzの周波数を発生する。２７MHz周波数の誤差範囲は、+/-３０ppm（２７０００００ +/- ８１０Hz）でなければならない。arrival time clock counter６２は、２７MHzの周波数のパルスをカウントするバイナリカウンタである。arrival_time_clock(i)は、時刻t(i)におけるarrival time clock counter６２のカウント値である。

smoothing buffer６４において、Rmaxは、スムージングバッファ６４がフルでないときのスムージングバッファ６４へのソースパケットの入力ビットレートである。スムージングバッファ６４がフルであるとき、スムージングバッファ６４への入力ビットレートは０である。

MPEG-2トランスポートストリームの出力タイミングを説明するに、現在のソースパケットのarrival_time_stampがarrival_time_clock(i)のLSB ３０ビットの値と等しいとき、そのソースパケットのトランスポートパケットは、スムージングバッファ６４から引き抜かれる。Rpkは、トランスポートパケットレートの瞬時的な最大値である。スムージングバッファ６４は、アンダーフローしてはならない。

DVR MPEG-2トランスポートストリームのプレーヤモデルのパラメータについては、上述したDVR MPEG-2トランスポートストリームのレコーダモデルのパラメータと同一である。

図９３は、Source packetのシンタクスを示す図である。transport_packet()は、ISO/IEC 13818-1で規定されるMPEG-2トランスポートパケットである。図９３に示したSource packetのシンタクス内のTP_extra_headerのシンタクスを図９４に示す。図９４に示したTP_extra_headerのシンタクスについて説明するに、copy_permission_indicatorは、トランスポートパケットのペイロードのコピー制限を表す整数である。コピー制限は、copy free、no more copy、copy once又はcopy prohibitedとすることができる。図９５は、copy_permission_indicatorの値と、それらによって指定されるモードの関係を示す。

copy_permission_indicatorは、全てのトランスポートパケットに付加される。IEEE1394デジタルインタフェースを使用して入力トランスポートストリームを記録する場合、copy_permission_indicatorの値は、IEEE1394 isochronous packet headerの中のEMI (Encryption Mode Indicator) の値に関連付けてもよい。IEEE1394デジタルインタフェースを使用しないで入力トランスポートストリームを記録する場合、copy_permission_indicatorの値は、トランスポートパケットの中に埋め込まれたCCIの値に関連付けてもよい。アナログ信号入力をセルフエンコードする場合、copy_permission_indicatorの値は、アナログ信号のCGMS-Aの値に関連付けてもよい。

arrival_time_stampは、次式

arrival_time_stamp(k) = arrival_time_clock(k)% 230

において、arrival_time_stampによって指定される値を持つ整数値である。

Clip AVストリームの定義をするに、Clip AVストリームは、上述したような定

義がされるDVR MPEG-2トランスポートストリームの構造を持たねばならない。arrival_time_clock(i)は、Clip AVストリームの中で連続して増加しなければならない。Clip AVストリームの中にシステムタイムベース（STCベース）の不連続点が存在したとしても、そのClip AVストリームのarrival_time_clock(i)は、連続して増加しなければならない。

Clip AVストリームの中の開始と終了の間のarrival_time_clock(i)の差分の最大値は、２６時間でなければならない。この制限は、MPEG-2トランスポートストリームの中にシステムタイムベース（STCベース）の不連続点が存在しない場合に、Clip AVストリームの中で同じ値のPTS（Presentation Time Stamp）が決して現れないことを保証する。MPEG2システムズ規格は、PTSのラップアラウンド周期を２３３／９００００秒（約２６．５時間）と規定している。

Bridge-Clip AVストリームの定義をするに、Bridge-Clip AVストリームは、上述したような定義がされるDVR MPEG-2トランスポートストリームの構造を持たねばならない。Bridge-Clip AVストリームは、１つのアライバルタイムベースの不連続点を含まなければならない。アライバルタイムベースの不連続点の前後のトランスポートストリームは、後述する符号化の制限に従わなければならず、かつ後述するDVR-STDに従わなければならない。

本実施の形態においては、編集におけるPlayItem間のビデオとオーディオのシームレス接続をサポートする。PlayItem間をシームレス接続にすることは、プレーヤ／レコーダに”データの連続供給”と”シームレスな復号処理”を保証する。”データの連続供給”とは、ファイルシステムが、デコーダにバッファのアンダーフローを起こさせることのないように必要なビットレートでデータを供給することを保証できることである。データのリアルタイム性を保証して、データをディスクから読み出すことができるように、データが十分な大きさの連続したブロック単位でストアされるようにする。

”シームレスな復号処理”とは、プレーヤが、デコーダの再生出力にポーズやギャップを起こさせることなく、ディスクに記録されたオーディオビデオデータを表示できることである。

シームレス接続されているPlayItemが参照するAVストリームについて説明する。

先行するPlayItemと現在のPlayItemの接続が、シームレス表示できるように保証されているかどうかは、現在のPlayItemにおいて定義されているConnection_Conditionフィールドから判断することができる。PlayItem間のシームレス接続は、Bridge-Clipを使用する方法と使用しない方法がある。

図９６は、Bridge-Clipを使用する場合の先行するPlayItemと現在のPlayItemの関係を示している。図９６においては、プレーヤが読み出すストリームデータが、影を付けて示されている。図９６に示したTS1は、Clip1（Clip AVストリーム）の影を付けられたストリームデータとBridge-ClipのRSPN_arrival_time_discontinuityより前の影を付けられたストリームデータから成る。

TS1のClip1の影を付けられたストリームデータは、先行するPlayItemのIN_time（図９６においてIN_time1で図示されている）に対応するプレゼンテーションユニットを復号するために必要なストリームのアドレスから、RSPN_exit_from_previous_Clipで参照されるソースパケットまでのストリームデータである。TS1に含まれるBridge-ClipのRSPN_arrival_time_discontinuityより前の影を付けられたストリームデータは、Bridge-Clipの最初のソースパケットから、RSPN_arrival_time_discontinuityで参照されるソースパケットの直前のソースパケットまでのストリームデータである。

また、図９６におけるTS2は、Clip2（Clip AVストリーム）の影を付けられたストリームデータとBridge-ClipのRSPN_arrival_time_discontinuity以後の影を付けられたストリームデータから成る。TS2に含まれるBridge-ClipのRSPN_arrival_time_discontinuity以後の影を付けられたストリームデータは、RSPN_arrival_time_discontinuityで参照されるソースパケットから、Bridge-Clipの最後のソースパケットまでのストリームデータである。TS2のClip2の影を付けられたストリームデータは、RSPN_enter_to_current_Clipで参照されるソースパケットから、現在のPlayItemのOUT_time（図９６においてOUT_time2で図示されている）に対応するプレゼンテーションユニットを復号するために必要なストリームのアドレスまでのストリームデータである。

図９７は、Bridge-Clipを使用しない場合の先行するPlayItemと現在のPlayItemの関係を示している。この場合、プレーヤが読み出すストリームデータは、影を

付けて示されている。図９７におけるTS1は、Clip1（Clip AVストリーム）の影を付けられたストリームデータから成る。TS1のClip1の影を付けられたストリームデータは、先行するPlayItemのIN_time（図９７においてIN_time1で図示されている）に対応するプレゼンテーションユニットを復号するために必要なストリームのアドレスから始まり、Clip1の最後のソースパケットまでのデータである。また、図９７におけるTS2は、Clip2（Clip AVストリーム）の影を付けられたストリームデータから成る。

TS2のClip2の影を付けられたストリームデータは、Clip2の最初のソースパケットから始まり、現在のPlayItemのOUT_time（図９７においてOUT_time2で図示されている）に対応するプレゼンテーションユニットを復号するために必要なストリームのアドレスまでのストリームデータである。

図９６と図９７において、TS1とT2は、ソースパケットの連続したストリームである。次に、TS1とTS2のストリーム規定と、それらの間の接続条件について考える。まず、シームレス接続のための符号化制限について考える。トランスポートストリームの符号化構造の制限として、まず、TS1とTS2の中に含まれるプログラムの数は、１でなければならない。TS1とTS2の中に含まれるビデオストリームの数は、１でなければならない。TS1とTS2の中に含まれるオーディオストリームの数は、２以下でなければならない。TS1とTS2の中に含まれるオーディオストリームの数は、等しくなければならない。TS1及び／又はTS2の中に、上記以外のエレメンタリストリーム又はプライベートストリームが含まれていてもよい。

ビデオビットストリームの制限について説明する。図９８は、ピクチャの表示順序で示すシームレス接続の例を示す図である。接続点においてビデオストリームをシームレスに表示できるためには、OUT_time1（Clip1のOUT_time）の後とIN_time2（Clip2のIN_time）の前に表示される不必要なピクチャは、接続点付近のClipの部分的なストリームを再エンコードするプロセスにより、除去されなければならない。

図９８に示したような場合において、BridgeSequenceを使用してシームレス接続を実現する例を、図９９に示す。RSPN_arrival_time_discontinuityより前のBridge-ClipのビデオストリームはClip1のOUT_time1に対応するピクチ

ャまでの符号化ビデオストリームから成る。そして、そのビデオストリームは先行するClip1のビデオストリームに接続され、１つの連続でMPEG2規格に従ったエレメンタリストリームとなるように再エンコードされている。

同様にして、RSPN_arrival_time_discontinuity以後のBridge-Clipのビデオストリームは、図９８のClip2のIN_time2に対応するピクチャ以後の符号化ビデオストリームから成る。そして、そのビデオストリームは、正しくデコード開始することができて、これに続くClip2のビデオストリームに接続され、１つの連続でMPEG2規格に従ったエレメンタリストリームとなるように再エンコードされている。Bridge-Clipを作るためには、一般に、数枚のピクチャは再エンコードしなければならず、それ以外のピクチャはオリジナルのClipからコピーすることができる。

図９８に示した例の場合にBridgeSequenceを使用しないでシームレス接続を実現する例を図１００に示す。Clip1のビデオストリームは、図９８のOUT_time1に対応するピクチャまでの符号化ビデオストリームから成り、それは、１つの連続でMPEG2規格に従ったエレメンタリストリームとなるように再エンコードされている。同様にして、Clip2のビデオストリームは、図９８のClip2のIN_time2に対応するピクチャ以後の符号化ビデオストリームから成り、それは、１つの連続でMPEG2規格に従ったエレメンタリストリームとなるように再エンコードされている。

ビデオストリームの符号化制限について説明するに、まず、TS1とTS2のビデオストリームのフレームレートは、等しくなければならない。TS1のビデオストリームは、sequence_end_codeで終端しなければならない。TS2のビデオストリームは、Sequence Header、GOP Header、そしてI-ピクチャで開始しなければならない。TS2のビデオストリームは、クローズドGOPで開始しなければならない。

ビットストリームの中で定義されるビデオプレゼンテーションユニット（フレーム又はフィールド）は、接続点を挟んで連続でなければならない。接続点において、フレーム又はフィールドのギャップがあってはならない。接続点において、トップ?ボトムのフィールドシーケンスは連続でなければならない。3-2プルダウンを使用するエンコードの場合は、”top_field_first”及び”repeat_first_field”フラグを書き換える必要があるかもしれない、又はフィールドギャップの発生を防ぐために局所的に再エンコードするようにしてもよい。

オーディオビットストリームの符号化制限について説明するに、TS1とTS2のオーディオのサンプリング周波数は、同じでなければならない。TS1とTS2のオーディオの符号化方法（例えば、MPEG1レイヤ2、AC-3、SESF LPCM、AAC）は、同じでなければならない。

次に、MPEG-2トランスポートストリームの符号化制限について説明するに、TS1のオーディオストリームの最後のオーディオフレームは、TS1の最後の表示ピクチャの表示終了時に等しい表示時刻を持つオーディオサンプルを含んでいなければならない。TS2のオーディオストリームの最初のオーディオフレームは、TS2の最初の表示ピクチャの表示開始時に等しい表示時刻を持つオーディオサンプルを含んでいなければならない。

接続点において、オーディオプレゼンテーションユニットのシーケンスにギャップがあってはならない。図１０１に示すように、２オーディオフレーム区間未満のオーディオプレゼンテーションユニットの長さで定義されるオーバーラップがあってもよい。TS2のエレメンタリストリームを伝送する最初のパケットは、ビデオパケットでなければならない。接続点におけるトランスポートストリームは、後述するDVR-STDに従わなくてはならない。

Clip及びBridge-Clipの制限について説明するに、TS1とTS2は、それぞれの中にアライバルタイムベースの不連続点を含んではならない。

以下の制限は、Bridge-Clipを使用する場合にのみ適用される。TS1の最後のソースパケットとTS2の最初のソースパケットの接続点においてのみ、Bridge-Clip AVストリームは、ただ１つのアライバルタイムベースの不連続点を持つ。ClipInfo()において定義されるRSPN_arrival_time_discontinuityが、その不連続点のアドレスを示し、それはTS2の最初のソースパケットを参照するアドレスを示さなければならない。

BridgeSequenceInfo()において定義されるRSPN_exit_from_previous_Clipによって参照されるソースパケットは、Clip1の中のどのソースパケットでもよい。それは、Aligned unitの境界である必要はない。BridgeSequenceInfo()において定義されるRSPN_enter_to_current_Clipによって参照されるソースパケットは、Clip2の中のどのソースパケットでもよい。それは、Aligned unitの境界である必要

はない。

PlayItemの制限について説明するに、先行するPlayItemのOUT_time（図９６、図９７において示されるOUT_time1）は、TS1の最後のビデオプレゼンテーションユニットの表示終了時刻を示さなければならない。現在のPlayItemのIN_time（F図９６、図９７において示されるIN_time2）は、TS2の最初のビデオプレゼンテーションユニットの表示開始時刻を示さなければならない。

Bridge-Clipを使用する場合のデータアロケーションの制限について、図１０２を参照して説明するに、シームレス接続は、ファイルシステムによってデータの連続供給が保証されるように作られなければならない。これは、Clip1（Clip AVストリームファイル）とClip2（Clip AVストリームファイル）に接続されるBridge-Clip AVストリームを、データアロケーション規定を満たすように配置することによって行われなければならない。

RSPN_exit_from_previous_Clip以前のClip1（Clip AVストリームファイル）のストリーム部分が、ハーフフラグメント以上の連続領域に配置されているように、RSPN_exit_from_previous_Clipが選択されなければならない。Bridge-Clip AVストリームのデータ長は、ハーフフラグメント以上の連続領域に配置されるように、選択されなければならない。RSPN_enter_to_current_Clip以後のClip2（Clip AVストリームファイル）のストリーム部分が、ハーフフラグメント以上の連続領域に配置されているように、RSPN_enter_to_current_Clipが選択されなければならない。

Bridge-Clipを使用しないでシームレス接続する場合のデータアロケーションの制限について、図１０３を参照して説明するに、シームレス接続は、ファイルシステムによってデータの連続供給が保証されるように作られなければならない。これは、Clip1（Clip AVストリームファイル）の最後の部分とClip2（Clip AVストリームファイル）の最初の部分を、データアロケーション規定を満たすように配置することによって行われなければならない。

Clip1（Clip AVストリームファイル）の最後のストリーム部分が、ハーフフラグメント以上の連続領域に配置されていなければならない。Clip2（Clip AVストリームファイル）の最初のストリーム部分が、ハーフフラグメント以上の連続領

域に配置されていなければならない。

次に、DVR-STDについて説明する。DVR-STDは、DVR MPEG-2トランスポートストリームの生成及び検証の際におけるデコード処理をモデル化するための概念モデルである。また、DVR-STDは、上述したシームレス接続された２つのPlayItemによって参照されるAVストリームの生成及び検証の際におけるデコード処理をモデル化するための概念モデルでもある。

DVR-STDモデルを図１０４に示す。図１０４に示したモデルには、DVR MPEG-2トランスポートストリームプレーヤモデルが構成要素として含まれている。n、TBn、MBn、EBn、TBsys、Bsys、Rxn、Rbxn、Rxsys、Dn、Dsys、On及びPn(k)の表記方法は、ISO/IEC13818-1のT-STDに定義されているものと同じである。すなわち、次の通りである。nは、エレメンタリストリームのインデックス番号である。TBnは、エレメンタリストリームnのトランスポートバッファである。

MBnは、エレメンタリストリームnの多重バッファである。ビデオストリームについてのみ存在する。EBnは、エレメンタリストリームnのエレメンタリストリームバッファである。ビデオストリームについてのみ存在する。TBsysは、復号中のプログラムのシステム情報のための入力バッファである。Bsysは、復号中のプログラムのシステム情報のためのシステムターゲットデコーダ内のメインバッファである。Rxnは、データがTBnから取り除かれる伝送レートである。Rbxnは、PESパケットペイロードがMBnから取り除かれる伝送レートである。ビデオストリームについてのみ存在する。

Rxsysは、データがTBsysから取り除かれる伝送レートである。Dnは、エレメンタリストリームnのデコーダである。Dsysは、復号中のプログラムのシステム情報に関するデコーダである。Onは、ビデオストリームnのre-ordering bufferである。Pn(k)は、エレメンタリストリームnのk番目のプレゼンテーションユニットである。

DVR-STDのデコーディングプロセスについて説明する。単一のDVR MPEG-2トランスポートストリームを再生している間は、トランスポートパケットをTB1、TBn又はTBsysのバッファへ入力するタイミングは、ソースパケットのarrival_time_stampにより決定される。TB1、MB1、EB1、TBn、Bn、TBsys及びBsysのバッファリング動作の規定は、ISO/IEC 13818-1に規定されているT-STDと同じである。復号動

作と表示動作の規定もまた、ISO/IEC 13818-1に規定されているT-STDと同じである。

シームレス接続されたPlayItemを再生している間のデコーディングプロセスについて説明する。ここでは、シームレス接続されたPlayItemによって参照される２つのAVストリームの再生について説明をすることにし、以後の説明では、上述した（例えば、図９６に示した）TS1とTS2の再生について説明する。TS1は、先行するストリームであり、TS2は、現在のストリームである。

図１０５は、あるAVストリーム（TS1）からそれにシームレスに接続された次のAVストリーム（TS2）へと移るときのトランスポートパケットの入力、復号、表示のタイミングチャートを示す。所定のAVストリーム（TS1）からそれにシームレスに接続された次のAVストリーム（TS2）へと移る間には、TS2のアライバルタイムベースの時間軸（図１０５においてATC2で示される）は、TS1のアライバルタイムベースの時間軸（図１０５においてATC1で示される）と同じでない。

また、TS2のシステムタイムベースの時間軸（図１０５においてSTC2で示される）は、TS1のシステムタイムベースの時間軸（図１０５においてSTC1で示される）と同じでない。ビデオの表示は、シームレスに連続していることが要求される。オーディオのプレゼンテーションユニットの表示時間にはオーバーラップがあってもよい。

DVR-STDへの入力タイミングについて説明する。時刻Ｔ1までの時間、すなわち、TS1の最後のビデオパケットがDVR-STDのTB1に入力終了するまでは、DVR-STDのTB1、TBn又はTBsysのバッファへの入力タイミングは、TS1のソースパケットのarrival_time_stampによって決定される。

TS1の残りのパケットは、TS_recording_rate(TS1)のビットレートでDVR-STDのTBn又はTBsysのバッファへ入力されなければならない。ここで、TS_recording_rate(TS1)は、Clip1に対応するClipInfo()において定義されるTS_recording_rateの値である。TS1の最後のバイトがバッファへ入力する時刻は、時刻Ｔ2である。したがって、時刻Ｔ1からＴ2までの区間では、ソースパケットのarrival_time_stampは無視される。

N1をTS1の最後のビデオパケットに続くTS1のトランスポートパケットのバイト

数とすると、時刻Ｔ1乃至Ｔ2までの時間DT1は、N1バイトがTS_recording_rate(TS1)のビットレートで入力終了するために必要な時間であり、次式により算出される。

DT1＝Ｔ2－Ｔ1＝N1 / TS_recording_rate (TS1)

時刻Ｔ1乃至Ｔ2までの間は、RXnとRXsysの値は共に、TS_recording_rate(TS1)の値に変化する。このルール以外のバッファリング動作は、T-STDと同じである。

Ｔ2の時刻において、arrival time clock counterは、TS2の最初のソースパケットのarrival_time_stampの値にリセットされる。DVR-STDのTB1、TBn又はTBsysのバッファへの入力タイミングは、TS2のソースパケットのarrival_time_stampによって決定される。RXnとRXsysは共に、T-STDにおいて定義されている値に変化する。

付加的なオーディオバッファリング及びシステムデータバッファリングについて説明するに、オーディオデコーダとシステムデコーダは、時刻Ｔ１からT2までの区間の入力データを処理することができるように、T-STDで定義されるバッファ量に加えて付加的なバッファ量（約１秒分のデータ量）が必要である。

ビデオのプレゼンテーションタイミングについて説明するに、ビデオプレゼンテーションユニットの表示は、接続点を通して、ギャップなしに連続でなければならない。ここで、STC1は、TS1のシステムタイムベースの時間軸（図１０５ではSTC1と図示されている）とし、STC2は、TS2のシステムタイムベースの時間軸（図９７ではSTC2と図示されている。正確には、STC2は、TS2の最初のPCRがT-STDに入力した時刻から開始する。）とする。

STC1とSTC2の間のオフセットは、次のように決定される。PTS1endは、TS1の最後のビデオプレゼンテーションユニットに対応するSTC1上のPTSであり、PTS2startは、TS2の最初のビデオプレゼンテーションユニットに対応するSTC2上のPTSであり、Tppは、TS1の最後のビデオプレゼンテーションユニットの表示期間とすると、２つのシステムタイムベースの間のオフセットSTC_deltaは、次式により算出される。

STC_delta = PTS1end + Tpp - PTS2start

オーディオのプレゼンテーションのタイミングについて説明するに、接続点に

おいて、オーディオプレゼンテーションユニットの表示タイミングのオーバーラップがあってもよく、それは０乃至２オーディオフレーム未満である（図１０５に図示されている"audio overlap"を参照）。どちらのオーディオサンプルを選択するかということと、オーディオプレゼンテーションユニットの表示を接続点の後の補正されたタイムベースに再同期することは、プレーヤ側により設定されることである。

DVR-STDのシステムタイムクロックについて説明するに、時刻Ｔ5において、TS1の最後のオーディオプレゼンテーションユニットが表示される。システムタイムクロックは、時刻Ｔ2からＴ5の間にオーバーラップしていてもよい。この区間では、DVR-STDは、システムタイムクロックを古いタイムベースの値（STC1）と新しいタイムベースの値（STC2）の間で切り換える。STC2の値は、次式により算出される。

STC2＝STC1－STC_delta

バッファリングの連続性について説明する。STC11video_endは、TS1の最後のビデオパケットの最後のバイトがDVR-STDのTB1へ到着するときのシステムタイムベースSTC1上のSTCの値である。STC22video_startは、TS2の最初のビデオパケットの最初のバイトがDVR-STDのTB1へ到着するときのシステムタイムベースSTC2上のSTCの値である。STC21video_endは、STC11video_endの値をシステムタイムベースSTC2上の値に換算した値である。STC21video_endは、次式により算出される。

STC21video_end = STC11video_end - STC_delta

DVR-STDに従うために、次の２つの条件を満たすことが要求される。まず、TS2の最初のビデオパケットのTB1への到着タイミングは、次に示す不等式を満たさなければならない。そして、次に示す不等式を満たさなければならない。

STC22video_start > STC21video_end + ΔT1

この不等式が満たされるように、Clip１及び／又はClip２の部分的なストリームを再エンコード及び／又は再多重化する必要がある場合は、その必要に応じて行われる。

次に、STC1とSTC2を同じ時間軸上に換算したシステムタイムベースの時間軸上において、TS1からのビデオパケットの入力とそれに続くTS2からのビデオパケッ

トの入力は、ビデオバッファをオーバーフロー及びアンダーフローさせてはならない。

このようなシンタクス、データ構造、規則に基づくことにより、記録媒体に記録されているデータの内容、再生情報などを適切に管理することができ、もって、ユーザが再生時に適切に記録媒体に記録されているデータの内容を確認したり、所望のデータを簡便に再生できるようにすることができる。

なお、本実施の形態は、多重化ストリームとしてMPEG-2トランスポートストリームを例にして説明しているが、これに限らず、MPEG2プログラムストリームや米国のDirecTVサービス（商標）で使用されているDSSトランスポートストリームについても適用することが可能である。

次に、mark_entry()及びrepresentative_picture__entry()のシンタクスが、図８１に示されるような構成である場合における、マーク点で示されるシーンの頭出し再生を行う場合の処理について、図１０６のフローチャートを参照して、説明する。

最初にステップＳ１において、記録再生装置１の制御部２３は、記録媒体１００から、DVRトランスポートストリームファイルのデータベースであるEP_map（図７０）、STC_Info（図５２）、Program_Info（図５４）及びClipMark（図７８）を読み出す。

ステップＳ２において、制御部２３は、ClipMark（図７８）のrepresentative_picture_entry（図８１）又はref_thumbnail_indexで参照されるピクチャからサムネイルのリストを作成し、ユーザインタフェース入出力としての端子２４から出力し、GUIのメニュー画面上に表示させる。この場合、ref_thumbnail_indexが有効な値を持つ場合、representative_picture_entryよりref_thumbnail_indexが優先される。

ステップＳ３において、ユーザが再生開始点のマーク点を指定する。これは、例えば、GUIとして表示されたメニュー画面上の中からユーザがサムネイル画像を選択することで行われる。制御部２３は、この選択操作に対応して、指定されたサムネイルに対応付けられているマーク点を取得する。

ステップＳ４において、制御部２３は、ステップＳ３で指定されたmark_entry

（図８１）のmark_Time_stampのPTSと、STC_sequence_idを取得する。

ステップＳ５において、制御部２３は、STC_Info（図５２）から、ステップＳ４で取得したSTC_sequence_idに対応するSTC時間軸が開始するソースパケット番号を取得する。

ステップＳ６において、制御部２３は、ステップＳ５で取得したSTC時間軸が開始するパケット番号と、ステップＳ４で取得したマーク点のPTSから、マーク点のPTSより時間的に前で、かつ、最も近いエントリポイント（Iピクチャ）のあるソースパケット番号を取得する。

ステップＳ７において、制御部２３は、ステップＳ６で取得したエントリポイントのあるソースパケット番号から、トランスポートストリームのデータを読み出し、AVデコーダ２７に供給させる。

ステップＳ８において、制御部２３は、AVデコーダ２７を制御し、ステップＳ４で取得したマーク点のPTSのピクチャから表示を開始させる。

以上の動作を、図１０７乃至１０９を参照してさらに説明する。

いま、図１０７に示されているように、DVRトランスポートストリームファイルは、STC_sequence_id=id０のSTC時間軸を有し、その時間軸が開始するソースパケット番号は、シーン開始点Ａのソースパケット番号より小さいものとする。そして、ソースパケット番号ＢからＣまでの間に、CM（コマーシャル）が挿入されているものとする。

このとき、図７０に示されるEP_mapに対応するEP_mapには、図１０８に示されるように、RSPN_EP_startで示されるＡ、Ｂ、Ｃに対応して、それぞれのPTSが、PTS_EP_startとして、PTS(A)、PTS(B)、PTS(C)として登録される。

また、図１０９に示されるように、図７８のClipMarkに対応するClipMarkには、図１０９に示されるように、シーンスタート、CMスタート及びCMエンドを表すマークタイプ（図７９）０x９２，０x９４，０x９５の値に対応して、mark_entryとrepresentative_picture_entryが記録される。

mark_entryのMark_Time_stampとしては、シーンスタート、CMスタート及びCMエンドに対応して、それぞれPTS(a1)、PTS(b0)、PTS(c0)が登録されており、それぞれのSTC_sequence_idは、いずれもid０とされている。

同様に、representative_picture_entryのMark_Time_stampとして、シーンスタート、CMスタート及びCMエンドに対応して、それぞれPTS(a2)、PTS(b0)、PTS(c0)が登録されており、それらはいずれもSTC_sequence_idが、id0とされている。

PTS(A)<PTS(a1)の場合、ステップＳ６において、パケット番号Ａが取得され、ステップＳ７において、パケット番号Ａから始まるトランスポートストリームが、AVデコーダ２７に供給され、ステップＳ８において、PTS(a1)のピクチャから表示が開始される。

次に、図１１０のフローチャートを参照して、mark_entryとrepresentative_picture_entryのシンタクスが、図８１に示されるような構成である場合におけるCMスキップ再生の処理について、図１１０のフローチャートを参照して説明する。

ステップＳ２１において、制御部２３は、EP_map（図７０）、STC_Info（図５２）、Program_Info（図５４）及びClipMark（図７８）を記録媒体１００から読み出す。ステップＳ２２において、ユーザは、ユーザインタフェース入出力としての端子２４からCMスキップ再生を指定する。

ステップＳ２３において、制御部２３は、マークタイプ（図７９）がCM開始点（０ｘ９４）であるマーク情報のPTSと、CM終了点（０ｘ９５）であるマーク情報のPTS、並びに対応するSTC_sequence_idを取得する（図８１）。

ステップＳ２４において、制御部２３は、STC_Info（図５２）からCM開始点と終了点の、STC_sequence_idに対応するSTC時間軸が開始するソースパケット番号を取得する。

ステップＳ２５において、制御部２３は、記録媒体１００からトランスポートストリームを読み出させ、それをAVデコーダ２７に供給し、デコードを開始させる。

ステップＳ２６において、制御部２３は、現在の表示画像がCM開始点のPTSの画像か否かを調べる。現在の表示画像がCM開始点のPTSの画像でない場合には、ステップＳ２７に進み、制御部２３は、画像の表示が継続される。その後、処理はステップＳ２５に戻り、それ以降の処理が繰り返し実行される。

ステップＳ２６において、現在の表示画像がCM開始点のPTSの画像であると判定された場合、ステップＳ２８に進み、制御部２３は、AVデコーダ２７を制御し、

デコード及び表示を停止させる。

次に、ステップＳ２９において、制御部２３は、CM終了点のSTC_sequence_idに対応するSTC時間軸が開始するパケット番号を取得し、そのパケット番号と、ステップＳ２３の処理で取得したCM終了点のPTSとから、その点のPTSより時間的に前で、かつ、最も近いエントリポイントのあるソースパケット番号を取得する。

ステップＳ３０において、制御部２３は、ステップＳ２９の処理で取得したエントリポイントのあるソースパケット番号から、トランスポートストリームのデータを読み出し、AVデコーダ２７に供給させる。

ステップＳ３１において、制御部２３は、AVデコーダ２７を制御し、CM終了点のPTSのピクチャから表示を再開させる。

図１０７乃至図１０９を参照して、以上の動作をさらに説明すると、CM開始点とCM終了点は、この例の場合、STC_sequence_id=id0という共通のSTC時間軸上に存在し、そのSTC時間軸が開始するソースパケット番号は、シーンの開始点のソースパケット番号Ａより小さいものとされている。

トランスポートストリームがデコードされ、ステップＳ２６で、表示時刻がPTS(b0)になったと判定された場合（CM開始点であると判定された場合）、AVデコーダ２７により表示が停止される。そして、PTS(C)＜PTS(c0)の場合、ステップＳ３０でパケット番号Ｃのデータから始まるストリームからデコードが再開され、ステップＳ３１において、PTS(c0)のピクチャから表示が再開される。

なお、この方法は、CMスキップ再生に限らず、一般的にClipMarkで指定される２点間のシーンをスキップして再生する場合にも、適用可能である。

次に、mark_entryとrepresentative_picture_entryが、図８２に示すシンタクス構造である場合における、マーク点で示されるCMの頭出し再生処理について、図１１２のフローチャートを参照して説明する。

ステップＳ４１において、制御部２３は、EP_map（図７０）、STC_Info（図５２）、Program_Info（図５４）及びClipMark（図７８）の情報を取得する。

次にステップＳ４２において、制御部２３は、ステップＳ４１で読み出したClipMark（図７８）に含まれるrepresentative_picture_entry（図８２）又はref_thumbnail_indexで参照されるピクチャからサムネイルのリストを生成し、GUIの

メニュー画面上に表示させる。ref_thumbnail_indexが有効な値を有する場合、representative_picture_entryよりref_thumbnail_indexが優先される。

ステップＳ４３において、ユーザは再生開始点のマーク点を指定する。この指定は、例えば、ステップＳ４２の処理で表示されたメニュー画面上の中から、ユーザがサムネイル画像を選択し、そのサムネイルに対応付けられているマーク点を指定することで行われる。

ステップＳ４４において、制御部２３は、ステップＳ４３の処理で指定されたマーク点のRSPN_ref_EP_startとoffset_num_pictures（図８２）を取得する。

ステップＳ４５において、制御部２３は、ステップＳ４４で取得したRSPN_ref_EP_startに対応するソースパケット番号からトランスポートストリームのデータを読み出し、AVデコーダ２７に供給させる。

ステップＳ４６において、制御部２３は、AVデコーダ２７を制御し、RSPN_ref_EP_startで参照されるピクチャから（表示はしないで）、表示すべきピクチャをカウントアップしていき、カウント値がoffset_num_picturesになったとき、そのピクチャから表示を開始させる。

以上の処理を、図１１３乃至図１１５を参照して、さらに説明する。この例においては、DVRトランスポートストリームファイルは、ソースパケット番号Ａからシーンが開始しており、ソースパケット番号Ｂからソースパケット ＣまでCMが挿入されている。このため、図１１４に示されるように、EP_mapには、RSPN_EP_startとしてのＡ、Ｂ、Ｃに対応して、PTS_EP_startとして、PTS(A)、PTS(B)、PTS(C)が登録されている。

また、図１１５に示されるように、シーンスタート、CMスタート及びCMエンドのマークタイプに対応して、mark_entryとrepresentative_picture_entryが登録されている。mark_entryには、シーンスタート、CMスタート及びCMエンドに対応して、RSPN_ref_EP_startとして、それぞれＡ、Ｂ、Ｃが登録され、offset_num_picturesとして、Ｍ１、Ｎ１、Ｎ２が登録されている。同様に、representative_picture_entryには、RSPN_ref_EP_startとして、シーンスタート、CMスタート及びCMエンドに対応して、それぞれＡ、Ｂ、Ｃが登録され、offset_num_picturesとして、Ｍ２、Ｎ１、Ｎ２がそれぞれ登録されている。

シーンスタートに当たるピクチャから頭出して再生が指令された場合、パケット番号Aのデータから始まるストリームからデコードが開始され、PTS(A)のピクチャから（表示をしないで）表示すべきピクチャをカウントアップをしていき、offset_num_picturesが、Ｍ１の値になったとき、そのピクチャから表示が開始される。

さらに、mark_entryとrepresentative_picture_entryのシンタクスが、図８２に示される構成である場合におけるCMスキップ再生の処理について、図１１６のフローチャートを参照して説明する。

ステップＳ６１において、制御部２３は、EP_map（図７０）、STC_Info（図５２）、Program_Info（図５４）及びClipMark（図７８）の情報を取得する。

ステップＳ６２において、ユーザがCMスキップ再生を指令すると、ステップＳ６３において、制御部２３は、マークタイプ（図７９）がCM開始点とCM終了点である各点のマーク情報として、RSPN_ref_EP_startとoffset_num_pictures（図８２）を取得する。そして、CM開始点のデータは、RSPN_ref_EP_start(1)、offset_num_pictures(1)とされ、CM終了点のデータは、RSPN_ref_EP_start(2)、offset_num_pictures(2)とされる。

ステップＳ６４において、制御部２３は、RSPN_ref_EP_start(1)、RSPN_ref_EP_start(2)に対応するPTSをEP_map（図７０）から取得する。

ステップＳ６５において、制御部２３は、トランスポートストリームを記録媒体１００から読み出させ、AVデコーダ２７に供給させる。

ステップＳ６６において、制御部２３は、現在の表示画像がRSPN_ref_EP_start(1)に対応するPTSのピクチャであるか否かを判定し、現在の表示画像がRSPN_ref_EP_start(1)に対応するPTSのピクチャでない場合には、ステップＳ６７に進み、ピクチャをそのまま継続的に表示させる。その後、処理はステップＳ６５に戻り、それ以降の処理が繰り返し実行される。

ステップＳ６６において、現在の表示画像がRSPN_ref_EP_start(1)に対応するPTSのピクチャであると判定された場合、ステップＳ６８に進み、制御部２３は、AVデコーダ２７を制御し、RSPN_ref_EP_start(1)に対応するPTSのピクチャから表示するピクチャをカウントアップしていき、カウント値がoffset_num_pictures

(1)になったとき、表示を停止させる。

ステップＳ６９において、制御部２３は、RSPN_ref_EP_start(2)のソースパケット番号からトランスポートストリームのデータを読み出し、AVデコーダ２７に供給させる。

ステップＳ７０において、制御部２３は、AVデコーダ２７を制御し、RSPN_ref_EP_start(2)に対応するPTSのピクチャから（表示をしないで）表示すべきピクチャをカウントアップしていき、カウント値がoffset_num_pictures(2)になったとき、そのピクチャから表示を開始させる。

以上の動作を、図１１３乃至図１１５を参照してさらに説明すると、まず、EP_map（図１１４）をもとに、パケット番号Ｂ，Ｃに対応する時刻PTS(B)，PTS(C)が得られる。そして、Clip AV streamがデコードされていき、表示時刻がPTS(B)になったとき、PTS(B)のピクチャから表示ピクチャがカウントアップされ、その値がＮ１（図１１５）になったとき、表示が停止される。

さらに、パケット番号Ｃのデータから始まるストリームからデコードが再開され、PTS(C)のピクチャから（表示をしないで）表示すべきピクチャをカウントアップしていき、その値がＮ２（図１１５）になったとき、そのピクチャから表示が再開される。

以上の処理は、CMスキップ再生に限らず、ClipMarkで指定された２点間のシーンをスキップさせて再生する場合にも、適用可能である。

次に、mark_entryとrepresentative_picture_entryのシンタクスが、図８４に示すような構成である場合における、マーク点で示されるシーンの頭出し再生処理について、図１１８のフローチャートを参照して説明する。

ステップＳ８１において、EP_map（図７０）、STC_Info（図５２）、Program_Info（図５４）、並びにClipMark（図７８）の情報が取得される。

ステップＳ８２において、制御部２３は、ClipMark（図７８）のrepresentative_picture_entry又はref_thumbnail_indexで参照されるピクチャからサムネイルのリストを生成し、GUIのメニュー画面として表示させる。ref_thumbnail_indexが有効な値を有する場合、representative_picture_entryよりref_thumbnail_indexが優先される。

ステップＳ８３において、ユーザは再生開始点のマーク点を指定する。この指定は、例えば、メニュー画面上の中からユーザがサムネイル画像を選択し、そのサムネイルに対応付けられているマーク点を指定することで行われる。

ステップＳ８４において、制御部２３は、ユーザから指定されたmark_entryのRSPN_mark_point（図８４）を取得する。

ステップＳ８５において、制御部２３は、マーク点のRSPN_mark_pointより前にあり、かつ、最も近いエントリポイントのソースパケット番号を、EP_map（図７０）から取得する。

ステップＳ８６において、制御部２３は、ステップＳ８５で取得したエントリポイントに対応するソースパケット番号からトランスポートストリームのデータを読み出し、AVデコーダ２７に供給させる。

ステップＳ８７において、制御部２３は、AVデコーダ２７を制御し、RSPN_mark_pointで参照されるピクチャから表示を開始させる。

以上の処理を、図１１９乃至図１２１を参照してさらに説明する。この例においては、DVRトランスポートストリームファイルが、ソースパケットＡでシーンスタートし、ソースパケット番号ＢからＣまでCMが挿入されている。このため、図１２０のEP_mapには、RSPN_EP_startとしてのＡ、Ｂ、Ｃに対応して、PTS_EP_startがそれぞれPTS(A)、PTS(B)、PTS(C)として登録されている。また、図１２１に示されるClipMarkに、シーンスタート、CMスタート及びCMエンドに対応して、markentryのRSPN_mark_pointとして、ａ１、ｂ１、ｃ１が、また、representative_picture_entryのRSPN_mark_pointとして、ａ２、ｂ１、ｃ１が、それぞれ登録されている。

シーンスタートにあたるピクチャから頭出して再生する場合、パケット番号Ａ＜ａ１とすると、パケット番号Ａのデータから始まるストリームからデコードが開始され、ソースパケット番号ａ１に対応するピクチャから表示が開始される。

次に、mark_entryとrepresentative_picture_entryのシンタクスが、図８４に示されるような構成である場合におけるCMスキップ再生の処理について、図１２２と図１２３のフローチャートを参照して説明する。

ステップＳ１０１において、制御部２３は、EP_map（図７０）、STC_Info（図

５２）、Program_Info（図５４）、並びにClipMark（図７０）の情報を取得する。

ステップＳ１０２において、ユーザは、CMスキップ再生を指定する。

ステップＳ１０３において、制御部２３は、マークタイプ（図７９）がCM開始点とCM終了点である各点のマーク情報のRSPN_mark_point（図８４）を取得する。そして、制御部２３は、CM開始点のデータをRSPN_mark_point（１）とし、CM終了点のデータをRSPN_mark_point（２）とする。

ステップＳ１０４において、制御部２３は、記録媒体１００からトランスポートストリームを読み出させ、AVデコーダ２７に出力し、デコードさせる。

ステップＳ１０５において、制御部２３は、現在の表示画像がRSPN_mark_point（１）に対応するピクチャであるか否かを判定し、現在の表示画像がRSPN_mark_point（１）に対応するピクチャでない場合には、ステップＳ１０６に進み、そのままピクチャを継続的に表示させる。その後、処理はステップＳ１０４に戻り、それ以降の処理が繰り返し実行される。

ステップＳ１０５において、現在の表示画像がRSPN_mark_point（１）に対応するピクチャであると判定された場合、ステップＳ１０７に進み、制御部２３はAVデコーダ２７を制御し、デコード及び表示を停止させる。

次に、ステップＳ１０８において、RSPN_mark_point（２）より前にあり、かつ、最も近いエントリポイントのあるソースパケット番号がEP_map（図７０）から取得される。

ステップＳ１０９において、制御部２３は、ステップＳ１０８で取得したエントリポイントに対応するソースパケット番号からトランスポートストリームのデータを読み出し、AVデコーダ２７に供給させる。

ステップＳ１１０において、制御部２３は、AVデコーダ２７を制御し、RSPN_mark_point（２）で参照されるピクチャから表示を再開させる。

以上の処理を図１１９乃至図１２１の例でさらに説明すると、Clip AV streamをデコードしていき、ソースパケット番号ｂ１（図１２１）に対応する表示ピクチャになったとき、表示が停止される。そして、ソースパケット番号Ｃ＜ソースパケット番号ｃ１とすると、パケット番号Ｃのデータから始まるストリームからデコードが再開され、ソースパケット番号ｃ１に対応するピクチャになったとき、

そのピクチャから表示が再開される。

以上のようにして、図１２４に示されるように、PlayList上で、タイムスタンプにより所定の位置を指定し、このタイムスタンプを各ClipのClip Informationにおいて、データアドレスに変換し、Clip AV streamの所定の位置にアクセスすることができる。

より具体的には、図１２５に示されるように、PlayList上において、PlayList Markとしてブックマークやリジューム点を、ユーザが時間軸上のタイムスタンプとして指定すると、そのPlayListは再生するとき、そのPlayListが参照しているClipのClipMarkを使用して、Clip AV streamのシーン開始点やシーン終了点にアクセスすることができる。

なお、ClipMarkのシンタクスは、図７８の例に代えて、図１２６に示すようにすることもできる。

この例においては、RSPN_markが、図７８のreserved_for_MakerID、mark_entry ()及びrepresetative_picture_entry ()に替えて挿入されている。このRSPN_markの３２ビットのフィールドは、AVストリームファイル上で、そのマークが参照するアクセスユニットの第１バイト目を含むソースパケットの相対アドレスを示す。RSPN_markは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClip Information fileにおいて定義され、offset_SPNの値を初期値としてカウントされる。

その他の構成は、図７８における場合と同様である。

ClipMarkのシンタクスは、さらに図１２７に示すように構成することもできる。この例においては、図１２６におけるRSPN_markの代わりに、RSPN_ref_EP_startとoffset_num_picturesが挿入されている。これらは、図８２に示した場合と同様のものである。

次に、図１２８は、ClipInfo()のシンタクスの別例を示す。

Clip_service_typeは、AVストリームファイルのタイプを示す。例えば、Clip_service_typeは、ビデオレコーディングやオーディオレコーディングなどのタイプを示す。また、例えば、Clip_service_typeは、デジタルＴＶ放送のプログラムが示すサービスタイプと同じ意味を持たせてもよい。例えば、日本のデジタルBS

放送の場合、サービスタイプは、テレビジョンサービス、音声サービス及びデータ放送サービスの３種類を持つ。AVストリームが含むプログラムのサービスタイプを代表する値をClip_service_typeにセットする。

transcode_mode_flagは、デジタル放送から受信されたMPEG-2トランスポートストリームの記録方法を示すフラグである。このフラグが１にセットされている場合、Clipに対応するAVストリームファイル中の少なくとも１つのエレメンタリストリームは再符号化されて記録されたことを示す。このフラグが１にセットされている場合、Clipに対応するAVストリームファイル中の全てのエレメンタリストリームはデジタル放送から受信されたままの内容で何も変更されないで記録されたことを示す。

その他のシンタクスフィールドは、図４６で説明した同名のフィールドと同じ意味を持つ。

次に、図１２９を参照して、ProgramInfo()の別例について説明する。

AVストリームファイルの中で本フォーマットが規定するところのプログラム内容が一定であるソースパケット列を、program-sequenceと呼ぶ。

AVストリームファイルの中で、新しいprogram-sequenceが開始するアドレスをProgramInfo()にストアする。このアドレスは、SPN_program_sequence_startにより示される。

AVストリームファイルの中にある最後のprogram-sequence以外のprogram-sequenceは、そのSPN_program_sequence_startで指されるソースパケットから開始し、その次のSPN_program_sequence_startで指されるソースパケットの直前のソースパケットで終了する。最後のprogram_sequenceは、そのSPN_program_sequence_startで指されるソースパケットから開始し、AVストリームファイルの最後のソースパケットで終了する。

program-sequenceは、STC-sequenceの境界をまたいでもよい。

lengthは、このlengthフィールドの直後のバイトからProgramInfo()の最後のバイトまでのバイト数を示す。

num_of_program_sequencesは、AVストリームファイルの中にあるprogram-sequenceの数を示す。

SPN_program_sequence_startは、AVストリームファイル上でprogram-sequenceが開始するアドレスを示す。SPN_program_sequence_startは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットから、０を初期値としてカウントされる。

ProgramInfo()の中でエントリされるSPN_program_sequence_startの値は、昇順に並んでいる。

SPN_program_sequence_startは、そのprogram_sequenceに対する最初のPMTを持つソースパケットを指していることを前提とする。SPN_program_sequence_startは、記録機（図１の記録再生装置１）がトランスポートストリーム中のPSI/SIを解析することによって作られる。記録機がPSI/SIを解析し、その変化を検出するまでの遅延時間が必要なために、SPN_program_sequence_startは、実際のPSI/SIの変化点から所定の時間以内にあるソースパケットを指してもよい。

program_map_PIDは、そのprogram-sequenceに適用できるPMT(program map table)を持つトランスポートパケットのPIDの値である。

num_of_streams_in_psは、そのprogram-sequenceの中で定義されるエレメンタリストリームの数を示す。

num_of_groupsは、そのprogram-sequenceの中で定義されるエレメンタリストリームのグループの数を示す。num_of_groupsは、１以上の値である。

トランスポートストリームのPSI/SIがエレメンタリストリームのグループ情報を持つ場合、num_of_groupsは、１以上の値をとることを想定している。それぞれのグループは、マルチビュープログラム中の１つのビューを構成する。

stream_PIDは、そのprogram-sequenceのprogram_map_PIDが参照するところのPMTの中で定義されているエレメンタリストリームに対するPIDの値を示す。

StreamCodingInfo()は、上記stream_PIDで指されるエレメンタリストリームの情報を示す。詳細は後述する。

num_of_streams_in_groupは、エレメンタリストリームのグループが持つエレメンタリストリームの数を示す。

stream_indexは、上記エレメンタリストリームのグループが持つエレメンタリストリームに対応するところの、シンタクス中のstream_indexのfor-loopで記述